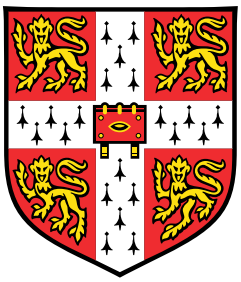
Probabilistic User Interface Design for Virtual and Augmented Reality Applications



John James Dudley

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Jesus College

September 2019

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

John James Dudley
September 2019

Probabilistic User Interface Design for Virtual and Augmented Reality Applications

John James Dudley

The central hypothesis of this thesis is that probabilistic user interface design provides an effective methodology for delivering productive and enjoyable applications in virtual reality (VR) and augmented reality (AR). This investigation is timely given the recent emergence of mass-market virtual and augmented reality head-mounted displays and growing demand for tailored applications and content. The design guidance for building compelling and productive applications for these environments is, however, currently lagging the pace at which the underlying technology is maturing. This is problematic given important differences between designing conventional 2D interfaces and interactions and their embodied 3D counterparts. This dissertation investigates probabilistic user interface design as a method for solving many of the novel challenges encountered when developing applications for VR and AR.

Probabilistic user interface design seeks to model the uncertain events in a system and identify, implement and validate strategies that drive improved system performance. This thesis addresses four research questions by applying a probabilistic treatment in four distinct but closely related case studies. These four case studies are selected to illustrate the flexibility and unique benefits offered by this method.

Research Question 1 asks how the probabilistic qualities of an interface can be determined and how this can inform design. This question is investigated in the context of text entry in VR with a probabilistic characterisation performed on two fundamental design choices. *Research Question 2* relates to the challenge of adapting AR applications to deployment contexts not knowable at design time. A study in which crowdworkers are employed to build a probabilistic understanding of the requirements for contextually adaptive AR answers this question. The text entry theme is revisited in answering *Research Questions 3* which asks how high levels of input noise can be mitigated through inference. A probabilistic text entry method specifically tailored for use in AR is implemented and evaluated. Finally, *Research Question 4* asks how the high dimensional design space in AR and VR applications can be efficiently explored to support ideal design choices. Interface refinement through probabilistic optimisation and crowdsourcing is shown to be highly efficient and effective for this purpose.

A probabilistic treatment in the design process has many potential benefits, principle among which is increased robustness to circumstances unanticipated at design time. This thesis contributes to the toolset and guidance available to designers and supports the development of next generation user interfaces specifically tailored to virtual and augmented reality.

Contributing Publications

Some of the work contained in this document has been published or is under submission. The contributing manuscripts are listed below.

- Dudley, J. J. and Kristensson, P. O. (2018). A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):8:1–8:37
  - *This journal paper contributes the section on representing uncertainty in Chapter 2.*
- Dudley, J. J., Benko, H., Wigdor, D., and Kristensson, P. O. (2019a). Performance Envelopes of Virtual Keyboard Text Input Strategies in Virtual Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality*, ISMAR 2019, pages 80–89. IEEEE
  - *This conference paper forms the basis of Chapter 4.*
- Wolf, D., Dudley, J. J., and Kristensson, P. O. (2018). Performance Envelopes of in-Air Direct and Smartwatch Indirect Control for Head-Mounted Augmented Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 347–354. IEEE
  - *This conference paper provides another example of a characterisation of augmented reality interactions and contributes conceptually to Chapter 4.*
- Dudley, J. J., Jacques, J. T., and Kristensson, P. O. (2020). Crowdsourcing Design Guidance for Augmented Reality: A Use Case in Contextually-Adaptive Text Content. *Under Submission*
  - *This journal paper forms the basis of Chapter 5.*
- Dudley, J. J., Schuff, H., and Kristensson, P. O. (2018a). Bare-Handed *3D* Drawing in Augmented Reality. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, pages 241–252, New York, NY, USA. ACM
  - *Portions of this conference paper contribute to Chapter 5.*
- Dudley, J. J., Vertanen, K., and Kristensson, P. O. (2018b). Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(6):30:1–30:40
  - *This journal paper forms the basis of Chapter 6.*
- Dudley, J. J., Jacques, J. T., and Kristensson, P. O. (2019b). Crowdsourcing Interface Feature Design with Bayesian Optimization. In *Proceedings of the 2019 Conference on Human Factors in Computing Systems*, CHI '19, pages 252:1–252:12, New York, NY, USA. ACM
  - *This conference paper forms the basis of Chapter 7.*

Acknowledgements

First I would like to thank my supervisor, Dr Per Ola Kristensson, who took a gamble on me. Without his initial encouragement and support I might never have started this journey. Certainly, without his energy and generous assistance I would never have finished. I am grateful not only for the knowledge and skills he has imparted but also for the clarity of thinking he has given to my approach to research.

Thank you to my fellow members of the Intelligent Interactive Systems Group: Bo, Jason, Max, Qisong, Sławomir and Stephen. There have been many enjoyable discussions both thesis and non-thesis related. I am especially indebted to Jason for his crowdsourcing expertise and assistance with Chapters 5 and 7 of this thesis. The wider Engineering Design Centre has also been a great source of friendly faces and helpful suggestions. Thank you in particular to Mari and Anna.

Although I made the tough decision to leave, I still carry the love for research and process that was instilled in me in the Smart Machines Group at UQ. It was certainly this foundation that inspired me to pursue a PhD. Thank you to Ross, Kev, Zane and Tyson for your example and continuing friendship.

I am grateful to the generous support I received from the Cambridge Trust, the Trimble Fund and Facebook Reality Labs during my PhD. Through these organisations too I had the opportunity to meet many amazing people who inspired aspects of my research.

Jesus College has been my family away from home. I have made many lifelong friends and will cherish many great memories. My life has been so much richer thanks to this great community and the JCBC and JCCC in particular.

Last, I would like to thank my family for their unending love and support. To my parents, you have been an inexhaustible source of encouragement and comfort. This milestone is merely one of many you have helped me to achieve and I dedicate this thesis to you. Finally, to Elise your patience has kept me calm and your expectations have kept me motivated. You have sacrificed as well on this journey and I owe you a debt I intend to repay over the rest of our lives together. If I ever need reminding, just show me this page.

To all, thank you again for the part you played and please know that I appreciate you.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Virtual and augmented reality head-mounted displays have recently entered the consumer market thanks in large part to significant advances in the underlying technology. These new devices support fundamentally new forms of work and leisure by embedding users in virtual or mixed virtual-physical environments. Such environments can deliver a powerful experience even when simply supporting the passive consumption of information. More powerful still are next-generation user interfaces in which this environment is both responsive and interactive. Delivering such next-generation user interfaces, however, is particularly challenging given the level of uncertainty inherent to this setting and its user interactions. The rapid advancement of the technology and its uptake by consumers has outpaced the development of formal design guidance and strategies for building these types of experiences. It is this user interface design challenge and its potential solution through a structured probabilistic treatment that is the focus of this thesis.

1.1 The Probabilistic User Interface

A probabilistic user interface recognises the uncertainty at its boundaries and actively seeks to accommodate or adapt in response. For example, conventional smartphone keyboards accommodate erroneous user input and provide word alternatives based on inference using the sentence context and the noisy input sequence. Users can then maintain high text entry rates even in challenging usage scenarios. This approach leverages the predictability of language and common user behaviours to deliver an enhanced user experience and elevate performance. Furthermore, it achieves this in a largely transparent and unobtrusive way.

Consider now the greatly expanded deployment scenarios and interaction spaces afforded by virtual and augmented reality. User interfaces developed for these environments are particularly exposed to uncertainty at their boundaries. Virtual reality (VR) places the user inside an

alternate simulated reality that is wholly computer generated. Nevertheless, the user remains in a physical setting that may be incorporated into the virtual environment, e.g. a virtual desk enhanced by the tactile experience afforded by a co-located physical desk. Augmented reality (AR) adds computer generated virtual content to the user's otherwise conventional physical reality. While there are many important distinctions between virtual and augmented reality (in particular, technical display and tracking considerations), there is sufficient commonality in the user interface design challenge for a unified examination. For conciseness, this thesis subsequently adopts the terminology introduced by Milgram and Kishino [122] defining the spectrum between an entirely virtual environment and the digitally augmented physical environment as Mixed Reality (MR). Although the interpretation of these different terms have evolved somewhat over the past two decades as they have entered popular parlance, the term Mixed Reality does serve to highlight the commonality over this continuum. Indeed, it is the blending of digital content and computer assisted interactions into the physical environment that offers the most exciting opportunities for extending the capabilities of the user and enhancing their perception.

There are various sources of uncertainty that differentiate MR from more conventional interaction environments. Three factors in particular make MR distinct from designing typical 2D interfaces: i) high levels of input noise; ii) high uncertainty in deployment contexts; and iii) high dimensionality in the design space.

The high levels of input noise encountered in MR are in part a consequence of embodied interactions involving large amplitude movements. Noise inherent in the motor control system is then often exacerbated by imprecise sensors frequently suffering from poor observability. Unlike conventional 2D interfaces where the touch point on a capacitive screen might, rightly or wrongly, be treated as an instantaneous and spatially confined interaction event, the continuous nature of typical 3D interactions frustrates attempts to collapse the spatial and temporal quality of events. The result is a high degree of ambiguity surrounding the user's intended input.

A significant obstacle to the future described earlier in this chapter where virtual and physical content is seamlessly blended together is the fact that designers cannot reliably envisage the range of contexts in which their MR applications will be deployed. This is in contrast to conventional mobile app development where the appearance of content and the experience constructed can be more reliably predicted. In most circumstances, it is a valid design assumption that the display characteristics and qualities of modern mobile devices (at least within the confines of the bezel) are largely consistent. Nevertheless, more advanced mobile applications may encounter the same types of challenges faced in MR as developers seek to incorporate contextual information, e.g. GPS location or direct feeds from the camera or microphone. Addressing the key problems faced in MR may therefore have cascading benefits

to more conventional technologies. The unique challenge for MR development, however, is that virtual content is added to the physical context compared with mobile development where contextual information is typically reproduced as virtual content. It is this unique aspect of MR that requires concerted research attention.

The high dimensionality of the MR design space is a product of the additional temporal and spatial freedom available to the user. The step from 2D to 3D might at first seem merely incremental but consider, as an example, how the regular discretisation of a cube yields n^3 units versus n^2 for the equivalent square in 2D. The spatial and temporal freedom afforded to the user too means that attention cannot, and should not, be confined to a prescribed region of focus (i.e. the $x \times y$ pixels of a conventional display). This high dimensionality of the design space is particularly challenging given the currently immature state of MR application development. As has occurred with the web and mobile devices, the interfaces and interactions that are known to be effective gradually emerge as convention. Without such established practice, it is difficult for the interaction designer to predict and/or accommodate the potentially complex and non-intuitive factor interactions which are likely to be encountered.

To highlight the challenges associated with designing next-generation interactive mixed reality applications, consider the following hypothetical design problem.

Hypothetical Design Problem

A developer is tasked with building an AR application for a construction company. The company wants to give their engineers performing building progress inspections an AR head-mounted display (HMD) that will allow them to identify and note any emerging or necessary divergence between the as-built and design specifications. The construction sites are distributed over large areas. Engineers will need to walk over the site, taking notes as they go. For safety reasons, engineers must not be encumbered by any hand-held input devices. Ongoing construction means that the site will be noisy. Furthermore, the notes taken will include numerous technical and company/site-specific terms. The use of voice commands and voice-to-text cannot, therefore, be the sole means of input.

Consider now three aspects of this task that are likely to be particularly troubling the developer in terms of the user interface design.

1. The HMD device provides only rudimentary hand and gesture based interaction with limited precision. How should the interface collect user input in circumstances where voice commands are not feasible?

2. Various engineers will need to use the application. They may obviously all have different arm lengths with different reach profiles. How should interface elements be positioned and structured to accommodate this variability?

3. The construction environment is constantly changing. How should virtual content be positioned and presented to ensure it remains visible and legible?

Although artificial, this design problem illustrates how many new considerations are introduced by the unique characteristics of AR user interfaces. A capable developer may of course deliver an ad hoc solution to this design challenge. The long term robustness of a solution that doesn't accommodate or respond to device, user or environmental uncertainty is, however, questionable. In recognition of this fact, there is anticipated value in a more formalised design approach that explicitly accommodates and models the inherent uncertainty. Probability is the mathematical approach used to represent uncertainty and there are a growing number of techniques suited to integration in a design process.

1.2 Probabilistic Approaches to Interaction Design

This thesis investigates strategies for mitigating and exploiting the probabilistic nature of virtual and augmented reality as it relates to interface and interaction design. At this early stage in the research of probabilistic user interfaces for mixed reality there is value in a broad investigation yielding an understanding of the most promising fundamental approaches and the scale and scope of the underlying research challenges in each category. This in contrast to a deep investigation of a single probabilistic technique or application area that might generate specialised insight but little general understanding of the wider approach.

Importantly too, this investigation examines a range of strategies that give coverage over several particularly troubling aspects of designing mixed reality applications. Specifically, this research explores how user performance and capabilities can be enhanced by: i) an improved understanding of the probabilistic nature of interaction events; ii) an improved understanding of the probabilistic nature of interface requirements; iii) the inference of intent from uncertain events; and iv) optimising interface features based on uncertain observations of performance. These four techniques can broadly be categorised as probabilistic treatments of user interface design. Each is described in more detail below:

- **Characterisation of interaction events:** Various factors contribute to high levels of user input noise in mixed reality. The additional mobility and interaction degrees of freedom supported means that users may be less precise. For virtual and augmented reality HMDs

incorporating hand tracking, sensor and tracking performance may introduce further noise. Characterising the uncertainty and distribution of interaction events is an important first step in probabilistic user interface design.

- **Characterisation of interface requirements:** The seamless overlay of virtual content on the physical world presents a difficult design problem given limited prior knowledge of deployment context. This challenge is well suited to a probabilistic treatment informed by a data driven understanding of the interface and user requirements. The appearance of virtual content may be dynamically adjusted in ways that respect various qualities such as designer specified aesthetics and smooth transitions.

- **Inferring intent from noisy input:** To accommodate noise in user input, intent may be inferred by leveraging a probabilistic understanding of the interaction space and task. This is a static and passive strategy in that the interface itself does not necessarily change in response. Rather, the application is engineered to accommodate noisy inputs by inferring original intent.

- **Interactive refinement of the design space:** The design of interfaces and interactions for mixed reality presents a complex multidimensional optimisation challenge. To address this challenge, the design problem may be reframed as a multi-user interactive refinement task. This strategy actively improves the interface according to the emerging performance model of the users.

This thesis posits that these four strategies represent the key pillars upon which productive and enjoyable next-generation MR interactive applications will be built. A probabilistic approach to interaction design is well suited to the evolving understanding and emerging design guidance in this space. In the short to mid term, these methods may be applied to deliver deterministic behaviour informed by an established probabilistic model. Such behaviour may suffer from inflexibility but is easier to comprehend, debug and deploy. As experience with these methods develops, it is likely that more advanced behaviours informed by continually evolving probabilistic models may gain traction. Such behaviour is more complex and less predictable but may deliver greater resilience. This thesis describes an investigation, that in part, serves to highlight both the current and future potential afforded by this design strategy. The framing and structure of this research investigation is addressed in the following section.

1.3 Central Hypothesis and Research Questions

The hypothetical design problem described in Section 1.1 can be distilled into the following general problem statement.

General Problem Statement

The unique challenges of noisy inputs, unknown deployment contexts and complex design spaces confounds the design of interactions and interfaces in mixed reality, resulting in a high risk that built solutions are inflexible and lack resilience. Such solutions are likely to deliver poor performance and user experiences.

This thesis argues that a probabilistic treatment using computational approaches is a viable means for solving this problem. The central research hypothesis is framed below.

Central Hypothesis

Probabilistic user interface design provides an effective methodology for delivering productive and enjoyable applications in mixed reality.

Dissecting this hypothesis with reference to the general problem statement, several research questions emerge.

Research Questions

1. How can a designer obtain an understanding of the probabilistic characteristics of an interface; and, how can this understanding inform design in mixed reality?
2. How can a data-driven probabilistic preference model for the appearance of virtual content in mixed reality be efficiently obtained; and, how can this be leveraged to enable adaptation of mixed reality applications to uncertain deployment contexts?
3. How can probabilistic inference be exploited to accommodate high levels of input noise in mixed reality applications to deliver more efficient interactions?
4. How can the unfamiliar and high dimensional design space for mixed reality applications be efficiently explored and refined through probabilistic optimisation?

The above research questions are addressed in the subsequent chapters of this thesis. A detailed outline, describing the correspondence between the research questions and chapters, is presented later in Section 1.4. The remainder of this section, however, outlines the perspective from which these questions are examined.

1.3.1 Key Terms

To ensure a common interpretation of the identified hypothesis and research questions, the key terms are defined below.

TERM	DEFINITION
Probabilistic User Interface	*A probabilistic user interface incorporates and exploits the estimated likelihood of events to deliver improved system performance and /or a better user experience. See Chapter 2 for a more detailed definition.*
Probabilistic User Interface Design	*Probabilistic user interface design seeks to model the uncertain events in the system and identify, implement and validate strategies that drive improved system performance. See Chapter 2 for a more detailed definition.*
Virtual Reality (VR)	*Virtual Reality places the user inside an alternate simulated reality that is wholly computer generated.*
Augmented Reality (AR)	*Augmented Reality adds computer generated virtual content to the user's otherwise conventional physical reality.*
Mixed Reality (MR)	*Mixed Reality is an overarching term describing the continuum between VR and AR. See [122].*
MR Application	*A user-targeted application supporting some computer-aided task in either Virtual or Augmented Reality. The user may play either a passive or active role.*
MR Interface	*The interface through which the user perceives or interacts with virtual content in a MR application.*
MR Interaction	*The means by which the user provides input to the MR interface.*
Input Noise	*Uncertainty introduced into interactions, chiefly through sensing inaccuracies and user imprecision.*
MR Deployment Context	*The physical environment in which the MR application is deployed.*

Probabilistic Characteristics	*The predictable features that describe the uncertainty of a given process. See Chapter 4 for a more detailed definition.*
Probabilistic Preference Model	*A model describing the relationship between the physical context of virtual content and the preferred visual representation of that content, accounting for uncertainty and variation in user preferences. See Chapter 5 for a more detailed definition.*
Probabilistic Inference	*The process by which the predictability of a given process is leveraged to make a reasoned estimate of the most likely intended input. See Chapter 6 for a more detailed definition.*
Probabilistic Optimisation	*The identification of ideal parameters for the design of an interface given uncertain estimates of the performance of the interface in the hands of users. See Chapter 7 for a more detailed definition.*

1.3.2 Research Objectives

A better understanding of the probabilistic treatments relevant to mixed reality user interfaces is critical for designers. Lacking in established heuristics and design patterns, developers of such applications require more support. The dearth of grounded frameworks to support the design of next-generation MR applications is what motivates this research project. Therefore, the primary research objective of this thesis is to advance and demonstrate the value of probabilistic approaches to MR interface design. The anticipated contribution to knowledge is a proven set of strategies that might be readily applied and expanded upon by designers.

In light of this goal, a conscious effort is made to promote coverage and relevance of the problem space explored. Therefore, two additional research objectives are relevant to the framing of this investigation. These are:

1. Demonstrate the concepts of probabilistic user interface design in a representative range of challenging and relevant mixed reality application scenarios.

2. Demonstrate a variety of complementary probabilistic approaches to highlight diversity in the strategies available.

1.3.3 Approach

This thesis attacks the above research questions through discrete but thematically related case studies (an overview is provided in Section 1.4). A common approach framed by the Design Research Methodology (DRM) [11] is applied despite the distinguishing features of each case study. Note that Chapter 3 presents a detailed explanation of the research methodology but the general approach is briefly outlined here.

The Research Clarification stage of the DRM framework provides an understanding of the broader design goals to be addressed using the probabilistic user interface design approach. A more concrete framing of these goals is provided by the specification of the four case studies as part of the Descriptive Study I stage of the DRM framework: each case study serves as a description of the existing situation in interface design for mixed reality and highlights the relevant design problems. The specific investigative process applied as part of the DRM's Prescriptive Study and Descriptive Study II stages is based on the concept of design principles: an aspect of the design that is a key determinant of user performance or experience. In each case, a prototype system is designed and developed with reference to these design principles to provide a platform for empirical investigation of the various probabilistic treatments. User performance and experience is then evaluated in controlled user studies. The empirical results, qualitative feedback and user observation inform the revision of these design principles and subsequent iteration of the system design. Finally, the efficacy of the given treatment and revised design principles are validated in a more realistic and less controlled application of the system.

1.4 Chapter Overview

The remaining chapters of this thesis are outlined below.

Chapter 2 reviews the existing literature and contextualises this investigation within the broader body of research. It explores the variety of existing computational approaches applied to probabilistic user interfaces. It also reviews the smaller, yet growing body of work which seeks to identify principles for the design of mixed reality applications. The literature and learnings derived from this research project ultimately inform the description of an emergent process for probabilistic user interface design. This process is presented in Section 2.3.

Chapter 3 describes the overarching methodology employed in the research project. Each of the subsequent chapters provides local specialisations of this methodology but there is a common framework applied in the four case studies. Chapter 3 explains and justifies this framework and motivates the case studies investigated.

Chapter 4 demonstrates the process of characterising the uncertainty in input events and their related behaviours (*Research Question 1*). A simulated probabilistic decoder is used to help explore the anticipated performance and behaviour characteristics relevant to text entry in virtual and augmented reality. An analysis of the captured data also informs the identification of the key levers of performance, informing subsequent stages of detailed design.

Chapter 5 explores a data-driven approach for dynamically adapting the user interface to the deployed physical context (*Research Question 2*). The specific challenge of adapting virtual content for visibility and legibility given background texture is investigated. A diverse and representative dataset is collected through crowdsourcing to build a model relating content colouration and placement given background texture to user preference.

Chapter 6 examines the inference of intent from noisy input in AR (*Research Question 3*). This investigation is made concrete by the development of a text entry method specifically designed for use with an AR HMD. A text entry method employing probabilistic decoding to mitigate noise in user inputs is developed and demonstrated. Chapter 6 takes particular care to highlight and exemplify the end-to-end process of probabilistic user interface design.

Chapter 7 investigates the potential for probabilistic optimisation to aid the design process given complex and unfamiliar design problems (*Research Question 4*). Bayesian optimisation is applied in concert with crowdsourced user testing to perform active online interface design refinement. Chapter 7 highlights the fact that Bayesian optimisation not only mitigates the effect of uncertainty in observations of user actions but also yields a model relating interface design features to expected performance.

Finally, Chapter 8 revisits the research questions and central hypothesis and reflects on the answers provided by this investigation. The key contributions of this research project are highlighted. Limitations and opportunities for future work are also identified.

Chapter 2

Probabilistic User Interface Design

The traditional approach to engineering design in the face of high uncertainty is to introduce large factors of safety. This approach can be preferable, and indeed more cost effective, when the anticipated loading cases on a system cannot be observed *a priori* or otherwise require a detailed investigation to accurately estimate. In most cases, a robust system that is 'over-engineered' but meets functional requirements is preferable to one that is prone to failure.

In the design of interactive systems, the uncertain loadings imposed by the user at the interface are often similarly difficult to anticipate. The parallel to the concept of a factor of safety is, however, difficult to articulate. The design of these types of systems therefore often resort to balanced trade-offs that deliver function within some tolerable bounds of expected user behaviour.

As an example, consider the design of a soft Qwerty keyboard implemented on an interactive surface. How should the keys and layout be sized? Potential analytical approaches to this design problem are to consult anthropometric data and/or conduct a Fitts' law experiment. Now consider the same keyboard deployed on a mobile platform subject to variable accelerations and high vibration, e.g. a terminal in a car, tank or aeroplane (e.g. Figure 2.1).[1] How should the key and layout sizing be adjusted? Increasing the size of the keys is inextricably coupled with increasing the size of the layout but there are likely also external constraints on the physical device size. Lacking any objective or analytical means for making such design choices, the system designer must balance these trade-offs to implement a solution that will, *probably* work for *most* people.

An alternative to this approach is the introduction of *intelligence* to the interactive system so that it can reconfigure the normally fixed mapping between inputs and outputs. This flexibility broadens the range of contexts and inputs for which the system will perform well. Such intelligence may be embedded through a variety of means from case based logic to a full

[1] Ignore, for the moment, the argument for alternative input strategies in such deployment environments.

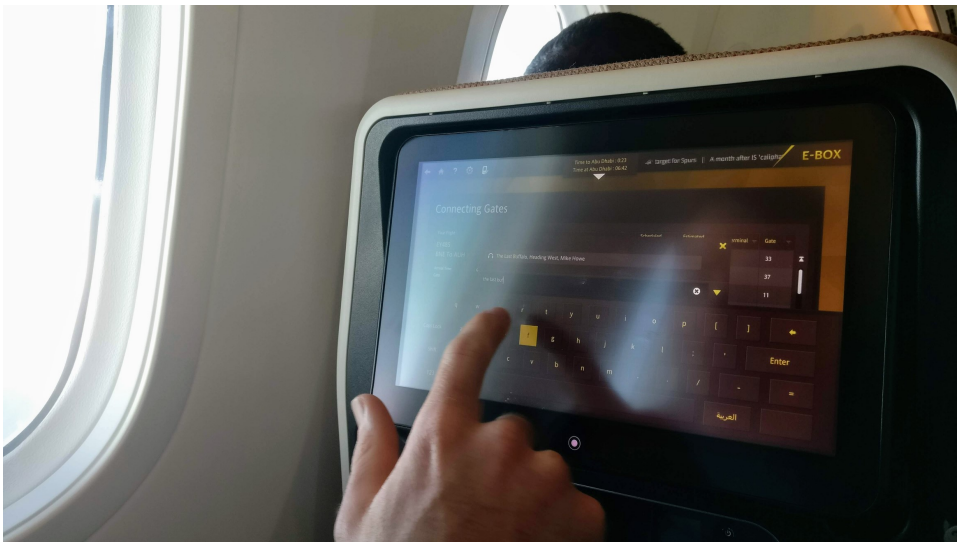


Fig. 2.1 A touch-based keyboard deployed in a commercial airline's entertainment system.

probabilistic treatment. The focus in this thesis is on the probabilistic treatment and design of the probabilistic user interface.

The argument for a probabilistic treatment over a programmatic one is compelling for four key reasons. First, it is rarely possible to reliably predict the range of possible contexts and behaviours of users. It is therefore difficult to ensure effective coverage with a finite collection of programmed logic. By contrast, a probabilistic treatment can provide for smooth transitions between system behaviours which may deliver better outcomes in unforeseen circumstances. Second, user interactions and expectations are rarely binary. That is, the user may themself be uncertain about the action they wish to perform and in many cases be satisfied by a range of possible responses from the system. An ability to represent and reflect this nuance helps to promote consistency with the user's mental model. Third, many forms of interactive systems deal with user interactions or data that follow predictable distributions. For example, the error in user touches often follows a Gaussian distribution while text input systems deal with language which typically follows a power law distribution. Exploiting and/or embedding awareness of these distributions in the system can deliver improved user experiences and performance. Fourth, there is great efficiency to be gained by exploiting regularities in the behaviour of users or the data with which they interact. A system can improve user productivity by reconfiguring its interface or by modifying its output when frequently observed events are encountered, e.g. an interface might be streamlined when a user begins a common set of actions, or a miss-typed word can be reliably auto-corrected.

Recognising the potential benefits of a probabilistic treatment is one thing but encoding this into a structured design process presents a significant conceptual challenge. It is this challenge and its specific relation to design in the context of mixed reality that is the focus of this chapter.

2.1 What is a Probabilistic User Interface?

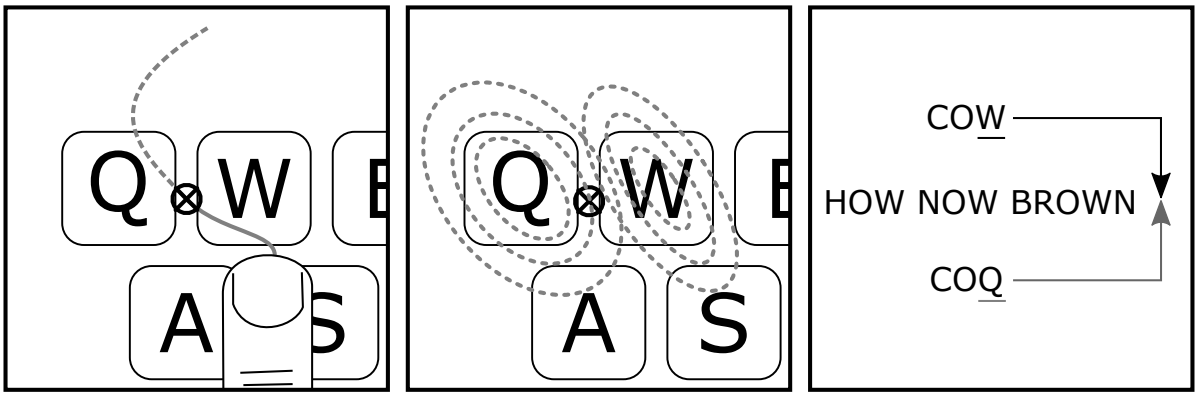
The interface is the portal connecting the user and the application. The user interacts with the interface to instruct the activity of the application and the application provides visibility of its behaviour to the user through the interface. It is therefore the user interface that establishes a productive bi-directional channel of communication and together these components produce an intelligent interactive system. The performance of this system as a whole is therefore a complex combination of the underlying performance of each component. Similarly, the user's experience with the system, i.e. how pleasing or productive they perceived their involvement to be, is a product of not only the interface and application qualities but also the user's background, expectations and biases among many other factors. As discussed already, often the actions of the user or the behaviours of the application are uncertain. Fortunately, there are often patterns to these uncertain events that can be represented and described through probability.

A probabilistic user interface incorporates and exploits the estimated likelihood of events to deliver improved system performance and/or user experience. The points in the system that benefit from a probabilistic treatment are those that introduce or exhibit high levels of uncertainty. Schwarz [162] describes how uncertainty may arise at three different stages in an interaction: sensing the input, interpreting the input, and selecting an appropriate action. To avoid the connotation that input uncertainty is solely a product of sensor noise, these three stages are reformulated here as: *input generation*, *input interpretation*, and *action determination*.

Definition: *Probabilistic User Interface*

A *probabilistic user interface* incorporates and exploits the estimated likelihood of events to deliver improved system performance and / or a better user experience.

As highlighted, sensing imprecision or inaccuracies may certainly introduce uncertainty at the point of input generation. Importantly, however, the user may themselves be imprecise or inaccurate when generating an input due to, for example, inherent error in the human motor control system or external forces that disturb regular motor control. Consider, as an example, the user generating a touch event on a mid-air virtualised keyboard as illustrated in Figure 2.2(a). Once an input is received, the interpretation of that input by the interactive system may introduce further uncertainty. For example, for the touch event illustrated in Figure 2.2(b),



a. b. c.

Fig. 2.2 An example of a user typing on a mid-air virtual keyboard to illustrate the distinct stages at which uncertainty is introduced: a) input generation, b) input interpretation, and c) action determination.

it is unclear whether this input should be interpreted as a *Q* or *W*. Last, there may be a degree of uncertainty in the determination by the system of what the appropriate output should be. As illustrated in Figure 2.2(c), the system must decide whether to update the displayed text to show *COW* or *COQ*.

With the aid of the hypothetical example presented in Figure 2.2, it is possible to highlight the potential benefit of a probabilistic treatment at each stage in this interaction. Imagine that the trajectory of the finger in the vicinity of the keyboard plane contains evidence of external disturbances such as a jolt from a footfall translated through to the arm. This evidence suggests a flattening and widening of the typical input distribution associated with touch events. At the interpretation stage, imagine that extensive user data collected indicates that the accuracy of touch events degrades as the distance of the target key from the centre of the keyboard increases. In other words, users are marginally more accurate in targeting *W* than *Q* such that a point equidistant between *W* and *Q* might reasonably be more likely to belong to *Q*. Finally, at action determination imagine that the system is aware of the prior context in the input field, e.g. *"HOW NOW BROWN CO"*. At this point, an awareness of the frequency distribution of words in English promotes *COW* over *COQ* and a multi-word language model deems *"HOW NOW BROWN COW"* considerably more likely than *"HOW NOW BROWN COQ"*.

The above example illustrates the advantages of a probabilistic user interface over an alternative that might otherwise take the user's input as *Q* since it is one pixel closer to the centre of *Q* than *W*. The three stages of uncertainty are now subjected to a more detailed examination contextualised by relevant research efforts addressing uncertainty at each stage.

2.1.1 Input Generation

Probabilistic user interfaces may address uncertainty at input generation via understanding the mechanism of the generating process or by modelling its observations. FingerCloud [151] demonstrates the concept of modulating uncertainty in the sensed position of the finger above a capacitive sensor array with a particle filter approach. This idea is extended by AnglePose [152] by also modelling the pose of the finger. Le et al. [97] present an interesting analysis of the potential classification of engaged fingers using low resolution capacitive touchscreen data. An awareness of the fingers used during input generation may aid subsequent disambiguation.

Rather than examining the process that generates touch events, Weir et al. [193] model touch inaccuracy over the surface of a mobile device. They find high-non-linearity and inter-user variation and show that user-specific models for touch offset correction outperform lumped user models. In recognition of the fact that particular user groups require specialised models, Montague et al. [123] introduce a framework for learning and distributing user models between devices and applications. Montague et al. [123] argue that these shared user models (SUM) are of particular value to user groups with visual and motor impairments, for whom there exists limited ability based modelling. The SUM framework enables dynamic adaptation of the interface according to a specific user's needs. This concept of incorporating both group and user-independent models of touch input has continued to receive attention in the literature. Mott and Wobbrock [124] demonstrate a strategy for mitigating errors in touch for motor impaired users or users in impairing situations. This strategy recognises that there is an unknown mapping between reported touch position and intended touch position and seeks to resolve this mapping.

Rather than modelling the input observations themselves as a means to understanding the generating process, Greis et al. [54] suggest inferring the user's uncertainty level from physiological sensor data and interaction behaviours. They investigated potential correlations between uncertainty and physiological and behavioural signals. An intuitive result of this investigation was that participants took longer to answer difficult questions and focused for more time on their answer prior to submission.

Fitt's law [43] and the Hick-Hyman law [64, 72] are two models describing human input behaviour which have been borrowed from psychology and have gained popularity in HCI. Regrettably, these empirical models do not capture or reflect the true variation exhibited both at the user and inter-user level. They serve as examples of how the aggregation of human performance data can mask the underlying distribution. Collapsing speed and accuracy, for example, into a single dimension inevitably results in information loss. Awareness of the characteristics of these distributions would arguable better inform the design of interactive systems.

2.1.2 Input Interpretation

The interpretation of user input can be particularly challenging as users may often be unintentionally inconsistent and imprecise. Other times uncertainty can arise due to laziness in articulation such as encountered with poor adherence to gesture templates. Added to this is the fact that systems must commonly accommodate multiple users with potentially widely varying interaction behaviours and goals.

The OOPS toolkit [113] is an attempt to give structure to the process of mediating input interpretation in instances where there is ambiguity. A sequence of inputs are maintained in a hierarchical graph and any resulting recognition nodes are considered ambiguous if they reflect one of several interpretations. Mankoff et al. [113] demonstrate how this common structure can be exploited to deliver different interaction techniques for resolving the ambiguity. Similarly, Quickset [25] describes a framework for integrated multimodal 'meaning fragments'. Fusing these separately sourced 'meanings' helps in resolving the intended user input. Bohus and Horvitz [12] learn a model to predict the probability of engagement with a conversational agent located in a communal office space. This offers an interesting example of how uncertain input interpretation might function differently depending on the expectations of the user.

In the gesture based text entry system Hex, Williamson and Murray-Smith [195] apply a clever trick to aid the interpretation of user intent. Tracing towards low probability entries is subject to higher resistance and conversely for highly probable entries, e.g. very little resistance in moving the trace cursor towards U when the previous selection was Q. Note that this approach turns the notion of addressing uncertainty in input interpretation on its head by forcing the user to be more explicit in their actions. Using a related strategy, both OctoPocus [9] and Hex [195] promote distinguishable gestures by presenting feedforward visualisations of canonical templates. Initially there may be several feasible gesture alternatives given the start point but as a particular gesture path is traced, these alternatives disappear or recede in prominence.

From an implementation point of view, various probabilistic techniques have been explored to represent the hypothesis space as it relates to input interpretation. XWand [196] utilises a dynamic Bayes network to intelligently resolve ambiguous targets and commands by also integrating speech and wand gestures. For example, pointing the wand at the light primes the command interpretation to expect either 'TurnOnLight' or 'TurnOffLight'. A subsequent user utterance of 'turn on' is thus easily disambiguated. Chai et al. [19] also use a graph representation to help capture and resolve uncertainty in multimodal interactions. Schwarz et al. [164] describe techniques based on a Monte Carlo approach to manage uncertainty in the assessment of state and for managing actions and feedback. Dumas et al. [39] suggest a fusion algorithm based on hidden Markov models (HMM). HMMs have also been used to deliver

probabilistic and adaptive keyboards [58, 59]. Uncertain<T> [13] is one example of an attempt at the programming level to capture and describe uncertainty as a pseudo data type.

2.1.3 Action Determination

At the point of action determination, uncertainty may be mitigated by exploiting models of user interaction goals or by intelligently inviting users to disambiguate the desired outcome. Schwarz et al. [163] provide several examples of how uncertainty can be propagated through the input interpretation and action determination stages and resolved before the action is finalised. The Lumière project [70] introduces the concept of Bayesian user models to embed interactive systems with the intelligence to infer user goals and needs. Horvitz et al. [70] highlight the fact that inferring desired system actions must consider the relative benefits and costs of those actions. For example, inferring an erroneous action that results in the closure of an application is perceived to be considerably more annoying than an action that corrects a desired literal string with a more likely auto-correction. Horvitz [69] encodes several of these key concepts in action determination under uncertainty with the description of mixed-initiative user interfaces. Horvitz [69] proposes 12 principles of mixed-initiative interfaces which not only touch on uncertainty but also aspects of timing and social awareness.

Liu et al. [102] describe an adaptive user interface that can learn individual user behaviours and styles. By keeping track of all user interaction events, the adaptive user interface can identify frequent event sequences. When a user subsequently embarks on a recognised sequence of interactions, the system can adjust to streamline the interface and reduce the number of interactions required.

The predictability of language can be exploited to intelligently facilitate the resolution of ambiguity in the user's desired outcome. Parakeet [184] is a speech recognition system designed for use with a mobile device to support rapid error corrections. Users are presented with a word confusion network constructed based on the likely alternative recognitions of a spoken phrase. The user can therefore easily modify the transcription action of the system. Also exploiting the predictability of language and an awareness of the error characteristics in input generation, VelociWatch [182] presents a detailed investigation of the error avoidance and correction strategies for a smartwatch keyboard.

BIGnav [103] demonstrates how the history of user interactions can be leveraged to inform the interpretation of task goals. Using a Bayesian experimental design approach, BIGnav maintains a model of the available information space and makes action determinations that will be most informative to the system. Through this approach, BIGnav is able to reduce the number of commands required to complete a multiscale pointing task. Extending this idea to file retrieval in a directory structure has also shown promise [104].

A key challenge in action determination is accommodating the user's subjective assessment of system performance. Humans are generally quite poor at decoupling a sequence of outcomes from the broader performance of a system. As Sheridan and Ferrell [167, pp. 37] note, "people readily adopt hypotheses about the nature and sources of probabilistic data; and these hypotheses, rather than the data, govern their behavior." A common manifestation of this is the gambler's fallacy. The counterpoint to this, however, is the fact that humans do possess a degree of intuition when it comes to the laws of proportions. As Pearl [144, pp. 15] notes, "For reasons of storage economy and generality we forget the actual experiences and retain their mental impressions in the forms of averages, weights, or (more vividly) abstract qualitative relationships that help us determine future actions."

2.2 What is Probabilistic User Interface Design?

The previous section has reviewed the concept of the probabilistic user interface and various efforts to exploit its power. Largely lacking, however, is any precise guidance on how such systems may be designed. Recognising the potential value of a probabilistic treatment of the user interface, it is important that system developers are provided with the tools that support their construction. These tools may be either high level design guidance or specific libraries or toolkits that offload much of the technical details.

Definition: *Probabilistic User Interface Design*

Probabilistic user interface design seeks to model the uncertain events in the system and identify, implement and validate strategies that drive improved system performance.

The principles of mixed-initiative user interfaces [69] already mentioned provide high level guidance that can be related to the more general form of probabilistic user interfaces. Schwarz [162] introduces the JULIA toolkit: a library developed to help streamline the task of building probabilistic interfaces. The JULIA toolkit explicitly treats user input as an uncertain process and rather than immediately executing actions, it evaluates the range of possible update operations to determine the most appropriate action. Developers can then specify interaction behaviour with the aid of probabilistic state machines. ProbUI [17] provides a tool for developers to deliver gesture interactions evaluated probabilistically. Using ProbUI, developers can define callbacks based on 'bounding behaviours' rather than 'bounding boxes'. Doherty et al. [32] specify user interactions as a stochastic system using process algebra.

The efforts of Horvitz [69], Schwarz [162], Buschek and Alt [17] and others provide an important foundation and reference point for intelligent interactive system designers. Still

lacking, however, is a generalisable design process that delivers structure to the how and why of various design choices.

2.2.1 Probabilistic Perspectives in HCI

The definition of a structured design process for probabilistic user interfaces in mixed reality is inevitably grounded in the more general theoretical perspectives relating to interactive system design. There is, of course, a degree of healthy tension and competition among different perspectives within HCI and so it is helpful to be explicit in defining the perspectives upon which this thesis is based.

Williamson [194] describes a process view of user inputs. From the system's perspective the user's inputs are noisy observations of some physical variable. These physical variables are themselves evidence for an underlying set of latent variables that reflect the actual intent of the user. Williamson [194, pp. 39] offers a concise summary of this process view and how interactions are formulated as a, "continuous control process, where the system is constantly engaged in recursively updating a distribution over the potential intentions of a user while feeding the result back at various timescales." Unfortunately, the control theory perspective on HCI breaks down when one considers how a user's goal state changes over time depending on an evolving understanding of the interface and in response to its behaviours. The concept of an inconsistent control reference generated from a non-deterministic underlying process is not handled well by the standard control theory parallel. In practice, the emergent behaviours of a system and a user working in coordination are difficult to characterise and model as a deterministic process without significant abstraction.

The control theory perspective on HCI has grown out of early efforts to derive human models of control (e.g. those of McRuer and Krendel [119]). Specifically, these models seek to describe how humans (much of the focus is on pilots) translate their sensory inputs into control actions. Doherty et al. [31] explore how these concepts can be applied to the design and analysis of interactive systems. In outlining the design process used to develop a control system for music generation by disabled users, they highlight the importance of examining the control characteristics of the user, device and controlled process. Williamson [194] also applies a control theory perspective to the task of interactive system design. As Williamson [194, pp. 21] notes, "It is feedback that transforms a simple one-directional communication into a control process." The role of the user under this perspective is to steer the system towards a goal state and reject disturbances once reached. For this to be effective, the interface must usefully (though not necessarily truly or comprehensively) represent the current state of the system.

Examining the general challenge of interface design from a task analysis perspective, Kirwan and Ainsworth [81] highlight the importance of a parallel consideration of the 'what' and 'how'. More specifically, there are two separate but related steps necessary to the design of effective interfaces: i) determining what information must be displayed to users in order to deliver an understanding of the current system state and required future state; and ii) determining how users will exercise their control through their actions.

The fact that in mixed reality the user is embedded within the virtual or mixed virtual-physical environment means that there is also great relevance of the theoretical perspectives promoted in the field of telepresence. Although focused on the challenges related to teleoperation of vehicles and manipulators, the characterisation by Sheridan [166] of the human-machine interactions over spatial and temporal distances is highly informative. Sheridan [166] describes how the closed control loop between human and machine unavoidably exhibits a degree of information loss at the interface boundary. This filtering effect occurs at both the mapping of commands from the user to the system (the efferent filter) and at the mapping between system action and user perception (the afferent filter). When acting on partial information, clearly uncertainty arises.

2.2.2 Communicating Uncertainty to the User

Uncertainty is an inevitable feature of data driven models in most real world applications. The concept of a probabilistic model and its limitations can be difficult to convey to the user and so many applications rely on simplified explanations. Most users are unlikely to comprehend the implications of working with a probabilistic model. User studies have found that even a single outlier in a classifier can result in significant confusion for users [78]. Users will calibrate their trust in the model both through individual predictions as well as the performance of the model as a whole [149]. Furthermore, intelligent user interfaces result in a co-adaptive process in which both the user and model will respond to the behaviour of the other. Establishing the right level of understanding among users and framing the interaction task appropriately is critical and non-trivial.

Non-experts unfamiliar with the internal behaviours of a computer program will construct their own mental model to aid their formulation of interaction strategies. This model will be derived in part from their past experience and knowledge. The adaptive interaction framework [143] further suggests that the strategy employed by a user will be dictated by their experience, their task level goals, and their ability to process information relevant to the task. From a user interaction perspective then, this third factor suggests a potential lever in terms of amplifying the cognitive ability of the user that might be activated to improve performance. While the mental model constructed does not have to be accurate, a poor model may have a highly detrimental

effect on user performance and thus, their perception of the effectiveness of the program [130]. It is perhaps useful to make the distinction between functional models that allow one to use a system versus structural models that allow one to comprehend how and why it works. Gillies et al. [51] argue that users should be aided in their construction of conceptual models in order to enhance their debugging capabilities. As Fogarty et al. [44] observe, evolution of the predictive model can result in seemingly unpredictable behaviour from the user's perspective. Kulesza et al. [94] investigate the impact that different explanations have on the fidelity of the mental models constructed by end-users. The results indicate that more detailed explanations about intelligent agents are useful if added understanding can be leveraged by the user to improve outcomes. Sarkar [160] proposes to exploit metamodels for confidence (is an output correct?), command (is the understanding complete?), and complexity (how simple was it to arrive at the output?) to augment machine learning models. Such metamodels would capture the information that is more intuitive and relevant for communication to end-users to support their understanding.

A number of strategies have been explored as a means to simplify the interpretation of model behaviour. ManiMatrix [76] allows users to interact directly with a classifier's confusion matrix and thereby steer classification behaviour. Ribeiro et al. [149] present explanations that are locally faithful representations of considerably more complex models. This approach supports interpretation while hiding the potentially confusing complexity underneath. Vidulin et al. [188], referencing constructivist learning theory, propose constraining the construction of decision trees to only represent relationships that are credible to the user. The use of exemplars to support understanding of classes appears to be a promising solution that resonates with users [78]. As a summative view of model quality, presenting best and worst matching samples has been shown to support more efficient model evaluation than ranking of the n best [44]. ReGroup, the social network group creation tool introduced by Amershi et al. [4], presents filters that were generated based on features in the model. Participants noted that these filters provided insight on the patterns that were being exploited by the model, and thus served the dual purpose of explaining the model as well as their intended function as an interaction element.

Uncertainty can be difficult to represent succinctly in a user interface. Sarkar et al. [161] demonstrated the potential for colouration to represent confidence within their BrainCel application, however, representing confidence through colouration in a speech recognition application [183] did not yield an improvement in user performance. Within the field of information visualisation, the representation of uncertainty is a key area of investigation. Vad et al. [179] describe a probabilistic interface for exploring a music library based on the mood classification of tracks. It conveys the uncertainty in the mood classification in its overview representation. In general, the objective of uncertainty visualisation is to provide representations that aid data

analysis and decision making [141]. It can also be useful to distinguish between different forms of uncertainty. Pang et al. [141] describe three types of uncertainty: statistical (distribution of the data), error (delta compared to datum) and range (interval of possible values).

Within the machine learning community there is also keen interest in representing model quality in ways that support human understanding. The technique known as t-Distributed Stochastic Neighbour Embedding (t-SNE) enables the visual representation of clustering models [180]. Such representations are easily queried and support non-expert reasoning on the level of confidence in the underlying model. Micallef et al. [120] present an investigation of explanatory methods for supporting Bayesian reasoning. The observations of this study reveal the difficulty of the design problem in that text *without* numbers paired with visual aids yielded higher performance than text *with* numbers and visuals.

The literature suggests that there is likely to be a close relationship between user tolerance of error and the level of clarity in system uncertainty [155]. The degree of error a user will tolerate in an application is task specific (e.g. compare an error encountered while withdrawing money from an ATM versus an erroneous turn instruction given by a navigation system [45]). If the user understands that they are in part responsible for an erroneous output, then they may be more forgiving in their perception of the system. Users will calibrate their trust of a system based on an understanding of the system properties. Muir and Moray [125] argue that behaviours must be observable for trust to grow.

2.2.3 Uncertainty in Virtual and Augmented Reality

This thesis explores the challenge of probabilistic user interface design for mixed reality. Although many of the approaches reviewed in this section translate across different interaction environments and deployments, there are several unique aspects of mixed reality that require targeted attention.

User interface design in augmented reality exposes several additional challenges relating to uncertainty. First, in contrast to conventional 2D interaction contexts, the virtualisation of the interface in 3D correspondingly expands the physical interaction space. Such embodied interactions are typically performed over larger magnitudes with less precision resulting in user input with high noise characteristics. For example, Arora et al. [5] investigated the impact of the lack of a physical surface on drawing inaccuracies. They demonstrated that providing additional visual guidance can substantially improve drawing accuracy. McGraw et al. [118] used an HTC Vive VR system with motion controls to allow users to control Hermite splines and create swept surfaces. Although not strictly an exploration in MR, Chen et al. [20] explore the potential for expanding the interaction space of a conventional mobile device above the screen.

Air+Touch [20] integrates both on-screen touch and above-screen gestures to dramatically expand the range and expressiveness of smartphone interactions.

Second, the extended interaction space means that the design space is also considerably broadened. This poses a significant challenge to designers as they seek to identify points in the high dimensional design space that meet their requirements.

Third, the embedding of virtual content within the environment creates a contextual coupling. However, the designer cannot know the range of deployment contexts at design time. An extension of this concept is the challenge of integrating different sources of interactions within the virtual and physical environments. Al-Sada et al. [2] offer an interesting investigation of the opportunities afforded by borrowing input from multiple connected devices to facilitate interaction with a HMD. Mapping input in different modalities and from different input devices represents a major challenge from the perspective of managing uncertainty.

These unique sources of uncertainty in mixed reality serve as useful exemplars throughout this thesis. They offer both a challenging and meaningful test of probabilistic user interface design.

2.3 An Emergent Design Process

The research efforts reviewed highlight the various benefits of a probabilistic treatment in interface design. Lacking, however, is a general design process for identifying, characterising, and addressing the various sources of uncertainty that may be encountered. This thesis seeks to bridge that gap in part by sketching out an emergent design process which may serve to complement existing software engineering practices. The investigation and specification of design practice is an active research field and there already exist well established processes in engineering (e.g. Pahl and Beitz [140]) and software (e.g. Sommerville [170]) design. The design process described here, therefore, is merely supplementary guidance that may be helpful when tackling an unfamiliar interaction task with unknown and uncertain interface and interaction qualities. More work is clearly required to justify and prove out this design process in practice, but as a non-core contribution of this thesis, this is considered beyond scope.

Note that an important conceptual distinction is made here between the description of a generic design process and the actual research methodology employed in this thesis. The former is framed according to the perspective of the MR application designer while the latter is the approach applied in answering the four research questions identified in Chapter 1. Chapter 3 presents the overarching research methodology where the focus is chiefly on building a system as a means to uncovering the generalisable design principles as they relate to the corresponding research questions. By contrast, the design process described in this section reflects the key

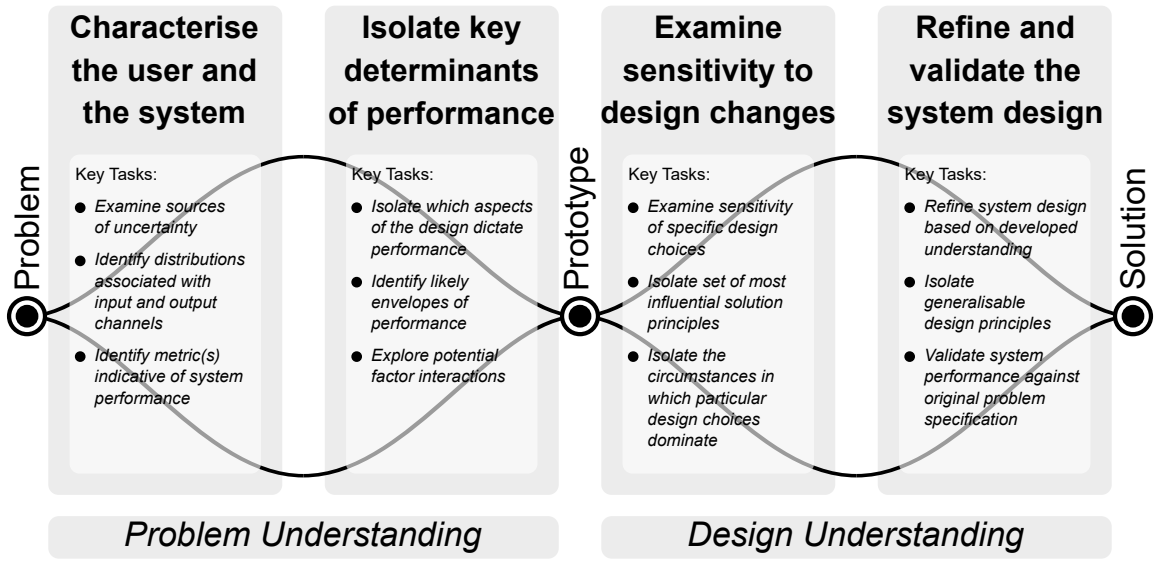


Fig. 2.3 An emergent design process for probabilistic user interface design.

stages which emerge (from the literature and the experience garnered through this research project) as being helpful for building a probabilistic interface from the point of view of the application developer. There is clearly a degree of overlap between the design process and the research methodology, particularly in terms of the focus on identifying the key levers of performance or design principles.

The emergent design process has four stages as illustrated in Figure 2.3. These design stages are loosely consistent with the divergent-convergent model of design. That is, there is a period of exploration of the problem before a concentration on the solution. The Design Council's double diamond[2] is a popular example of this model. Like the Design Council's double diamond, the four stages identified also describe two instances of divergence and convergence. The process described in Figure 2.3 assumes that the problem to be addressed has already been identified and a valid need exists. This represents an important distinction from the Design Council's model in which the initial divergent-convergent phase relates to developing an understanding of the need. In practice, an objective understanding of the need is important in determining when the probabilistic user interface design approach may be appropriate (see Section 8.5 for more discussion of this point). For completeness too, it is important to highlight how understanding subsequent downstream activities (post solution specification) might influence design choices, e.g. deployment, maintenance and decommissioning. In the context of this thesis, however, the focus is primarily on the core design stages and so Figure 2.3 serves as a good model for the

[2]See https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond

concepts demonstrated. The four stages are now described in further detail. Examples of their application in the subsequent chapters of this thesis are also provided.

2.3.1 Characterise the User and the System

As highlighted by Doherty et al. [31], characterising the user and the system is an important place to start. This stage involves examining the sources of uncertainty and their anticipated distributions. As reflected by Figure 2.3, this stage is typically divergent as the range of factors relevant to user and system performance may not be known *a priori* which necessitates exploration. This investigation also helps inform the selection of the metric or metrics of performance that can be evaluated throughout the design process to measure factor influences. It may also be possible to characterise high level system performance distributions without extensive development effort by simulating system behaviours or through Wizard of Oz methods.

This stage is exemplified in Chapter 4 where a simulated text input decoder is used to capture data on the error and behavioural characteristics of users in a novel VR typing setting. Characterisation is also performed in Chapter 5 through an efficient crowdsourcing method to generate an understanding of the range of background contexts to be expected in AR deployments. Although not presented in this thesis, Wolf et al. [197] provide an informative example of a detailed characterisation of touch versus smartwatch based selection in AR.

2.3.2 Isolate Key Determinants of Performance

The next stage in the process is to isolate which aspects of the design actually dictate summative system performance. The outcomes of the previous stage, while interesting, do not necessarily indicate where design efforts should be targeted. Using the outcomes of stage one, however, it is often possible to determine, either experimentally or analytically, the dominant levers of system performance. Figure 2.3 illustrates that this stage is convergent in that a broad spectrum of point observations of user and system performance are distilled into their underlying causes. Here the design focus is largely on the 'what' described by Kirwan and Ainsworth [81]. The outcome of this stage is an improved appreciation of the design problem and insight into how various design decisions might be expected to deliver certain performance levels. This informs an understanding of which design solutions should be prototyped and promoted for further examination.

In addressing *Research Question 1*, Chapter 4 illustrates how a probabilistic characterisation of different text input strategies informs the identification of which design choices (and to what extent) dominate net typing performance. Similarly, analysis of the crowdsourced datasets collected in Chapter 5 identifies which qualities of the background context have the greatest

influence on user preference. Chapter 6 also provides an example of how analysing the low level metric of inter-key interval highlights significant entry rate potential for the probabilistic touch-driven text input method examined.

2.3.3 Examine Sensitivity to Design Changes

With a narrower selection of suitable design solutions, it is now possible to develop a prototype system with which the sensitivity of more specific design choices can be examined. This may typically involve experimenting with different feedback and control strategies as reflected by the divergence shown in Figure 2.3. Here the design focus is largely on the 'how' described by Kirwan and Ainsworth [81]. From this investigation, there emerges an appreciation of which solutions are actually likely to deliver improvements and under what circumstances.

This stage is most clearly illustrated in Chapters 6 and 7. Chapter 6 evaluates a range of design choices identified to be influential to the performance of text entry in AR. For example, the effect of providing a literal fall-back method, occlusion reduction strategies and word predictions are evaluated in a series of controlled user studies. A fundamentally different approach is presented in Chapter 7 where a range of design parameters are efficiently evaluated using crowdsourcing. In both cases, the examination yields insight into the effect of various specific design decisions.

2.3.4 Refine and Validate the System Design

In this final stage of the design process, the system is refined based on an improved awareness of the impact of the various design choices. Additional insights revealed for enhancing performance are applied and generalisable design principles are identified. This is followed by the validation of system performance in light of the original problem formulation. This may necessitate user experiments with the system to confirm the satisfaction of key design objectives. The outcome is a functional probabilistic user interface solution.

This final refinement and validation stage is briefly illustrated in Chapters 6 and 7. It is important to recall, however, the distinction discussed above between this design process and the research methodology applied in the thesis, i.e. the objective in this project is to answer the research questions and the systems developed serve as vehicles for this purpose rather than fully formed solutions in their own right. The VISAR keyboard described in Chapter 6 is refined and validated in a simple study exploring how users can be supported to perform spatially distributed text entry tasks. Chapter 7 concludes with a design case study intended to illustrate how the interface refinement procedure might be applied to a novel and unfamiliar design problem.

2.4 Summary

This chapter has examined the broader challenges of designing and developing a probabilistic interface. The probabilistic user interface design approach is motivated by the highlighted benefits of a probabilistic treatment in interactive system design. These are, i) flexibility in the face of unseen cases, ii) alignment with the non-binary qualities of user interactions and expectations, and iii) relatability to commonly observed input and data distributions in HCI. A definition of the probabilistic user interface is offered with several illustrative examples. The three stages in an interaction that introduce uncertainty are discussed with reference to related research efforts that specifically target these stages with mitigation strategies. Finally, efforts to tackle both the general and specific design problem are examined. This in turn informs the four-part high level design process proposed. Different stages of this design process are demonstrated in the remainder of this thesis as part of answering the four research questions identified in Chapter 1.

Chapter 3

Research Methodology

This chapter describes the overarching methodological framework as well as the specific investigative process applied in this thesis. The Design Research Methodology (DRM) [11] guides the research strategy in high-level terms. The DRM framework is used to elucidate the current situation and opportunities for design support. A key outcome of the initial stages of this methodology is the specification of four case studies as concrete targets for investigation.

At the case study level, the specific investigative process applied is based on the idea of learning through building. In each of the four case studies, a prototype system is designed, built and evaluated in user testing. This strategy supports the extraction of key design related information from two perspectives: as a designer and as a user. Furthermore, the prototype systems serve as a means for validating that the probabilistic treatments proposed are both feasible and effective. The six stages of the investigative process are outlined later in this chapter. Finally, the chapter concludes with a discussion of several important limitations applied in scoping the investigation as well as a critical review of the choice of the specific case studies in the context of the secondary research objectives stated in Chapter 1.

3.1 Overarching Framework

To understand the research methodology applied in this thesis, it is useful to first revisit the Central Hypothesis.

Central Hypothesis

Probabilistic user interface design provides an effective methodology for delivering productive and enjoyable applications in mixed reality.

A critical decomposition of this hypothesis suggests three key objectives: i) understanding what aspects of mixed reality interfaces are suitable for this treatment; ii) developing a system that makes the treatment testable; and iii) assessing whether an improvement has been delivered. The Design Research Methodology offers a structured framework for tackling these three objectives. It provides guidance on the identification of the core research goals and a roadmap for determining appropriate design interventions followed by their implementation and evaluation.

The four stages of the DRM framework are Research Clarification, Descriptive Study I, Prescriptive Study, and Descriptive Study II. These four stages as they relate to the specific activities and outcomes of this thesis are summarised in the following table.

STAGE		ACTIVITY	OUTCOMES
1	**Research Clarification**	*Formulate research goals through analysis of literature.*	• Domain Literature Review • General Problem Statement • Central Hypothesis • Research Questions
2	**Descriptive Study I**	*Establish targets of investigation and define existing situation, highlighting relevant design problems.*	• Case Study Specification • Problem Specific Literature Review • Preliminary Design Principles
3	**Prescriptive Study**	*Develop support to address identified design problems.*	• Prototype System • Evaluation Plan
4	**Descriptive Study II**	*Evaluate effectiveness of the support in the context of its intended application.*	• Empirical Evaluation • Validated Design Principles

The aim in the Research Clarification stage is to identify the goals and structure of the overall research project, e.g. focus, hypothesis and research questions. The outcomes of this stage have already been presented in Chapters 1 and 2. The research questions are periodically revisited throughout the thesis to ensure alignment with the project goals.

The Descriptive Study I (DS-I) stage seeks to deliver greater understanding around the specific design problem by assessing the current state and opportunities for improvement. In the context of this study, four distinct design challenges are examined through a case study approach. The factors motivating the choice of these particular case studies are discussed later in Section 3.5. Chapters 4 to 7 present the four case studies and each chapter offers a literature review-based characterisation of the specific design problem as well as the opportunities

afforded by the probabilistic user interface design approach. The problem specific literature review also suggests a preliminary set of design principles (factors which are hypothesised to dictate user performance and experience) to explore in the Prescriptive Study and Descriptive Study II stages.

The Prescriptive Study (PS) stage involves the implementation of a support to address the identified design problem. In each case study a prototype system is developed according to the principles of probabilistic user interface design with the express purpose of conducting user evaluations to assess the effectiveness of the support. This evaluation is performed as part of the Descriptive Study II (DS-II) stage yielding a quantified assessment of the merits of the proposed solution. A secondary outcome of this stage as a validation of the design principles theorised to be influential to the performance of the target system.

In terms of the various types of research project identified by Blessing and Chakrabarti [11, pp. 62], this thesis serves as an example of Type 6: development of support and comprehensive evaluation. This categorisation reflects the fact that three of the four case studies presented offer an empirical evaluation of the prototype systems developed. The choice of well constrained case studies and controllable design interventions made this feasible within the time frame of the research project. A concentrated focus was also maintained throughout by applying a structured investigative process as outlined in the following section.

3.2 Investigative Process

The investigative process applied in this research project is based on the assumption that there exists a distinct set of design principles that dictate user performance and experience for a given mixed reality interface. This perspective is based on the conception of HCI as an engineering discipline [109]. These design principles are often not immediately apparent, especially when the interface or design approach is particularly novel. It is, however, usually possible to derive a preliminary set of principles by examination of related work and through early pilot studies. As Long and Dowell [109] note, the obvious objection to approaching HCI from this perspective is the indeterministic nature of human behaviour. Nevertheless, within certain bounds the behaviour or at least range of behaviours of humans can be presumed. Indeed, this is a strong argument for modelling and accommodating the stochastic qualities of the user. The risks of this strategy are outweighed by the major advantage of this approach in that validated design principles become operational and generalisable.

Each case study in this thesis therefore seeks to identify the design principles relevant to the target system and how they specifically influence the user's performance and experience. This part of the methodology is chiefly empirical with system evaluations conducted through

user studies. The six stages of the investigative process outlined below take influence from the engineering design process, and specifically the formulations described by Pahl and Beitz [140] and Samuel and Weir [158]. An important distinction from standard engineering design, however, is that the design outcome is secondary to the understanding developed around the relevant design principles. It is important to highlight that there may be a degree of iterative refinement manifesting as a feedback loop between stages 4 and 5.

STAGE		ACTIVITY
1	**Describe overall system function**	*Identify how the application should perform and behave in an ideal implementation.*
2	**Compile preliminary design principles**	*Compile a preliminary set of design principles through review of the related work and pilot experimentation.*
3	**Build test application**	*Develop a test application that encapsulates the identified design principles and, to the extent possible, enables isolation of the influence of each principle on user performance and experience.*
4	**Perform user testing**	*Examine performance and experience through user testing with the developed application.*
5	**Refine design principles**	*Based on the outcomes of the user testing, update and refine the list of design principles.*
6	**Validate refined application**	*Revise the design of the application based on the updated principles and validate its utility in a representative test scenario.*

The four case studies presented in this thesis each involve slight local modification of this process. Most significantly, in Chapter 7 the application examined is a design support tool rather than a mixed reality application per se. Therefore, the design principles relate more to the process of refining interface features than to the actual design of the features themselves. Chapters 4, 5 and 6 are more typical studies of mixed reality interface designs directly relatable to individual user experiences and performance. Where appropriate, local specialisations of the above methodology are introduced in each chapter.

The six stages of this investigative process are described in more detail below. Where relevant, pertinent examples illustrating the application of these stages are drawn from the studies described in this thesis.

Stage 1. Describe Overall System Function

The first stage in the process involves describing the desirable behaviour of the system. This

may be trivial, such as in the case of the text entry systems examined in Chapters 4 and 6, where the desired overall function is to translate user input into text as rapidly and accurately as possible. In practice, the means by which this behaviour is achieved is often less obvious. The description of overall system function is ideally framed to be solution neutral to avoid bias towards any particular solution early on in the process.

Stage 2. Compile Preliminary Design Principles

With the desired system functions identified, the next stage involves compiling a preliminary set of design principles. That is, principles which are known to dictate good or bad performance. These design principles may be obtained through review of related literature or through short pilot studies. For example, in the context of the text entry system examined in Chapter 6, a design principle relevant to heavily rate limited input is widely recognised to be the use of word completions and suggestions.

Stage 3. Build Test Application

At stage 3 a prototype system is built with reference to the compiled design principles. Ideally this system not only embodies the design principles but also facilitates their isolated investigation. For example, in Chapter 4, a hypothesised key design principle is the provision of a physical surface aligned to the virtual keyboard plane for productive text entry. In this case study, the system was built to enable independent testing of this particular design choice.

Stage 4. Perform User Testing

The built system is now subjected to controlled user testing. These experiments examine the validity and sensitivity of the identified design principles. All four case studies in this thesis involved user experiments, with the experiments of Chapters 4 and 6 performed in the lab and Chapters 5 and 7 performed chiefly with crowdworkers.

Stage 5. Refine Design Principles

Based on the outcomes of the user testing, the set of identified design principles is refined at stage 5. As described above, there is typically a degree of iteration between stages 4 and 5. For example, the design process pursued in Chapter 6 describes how the data obtained from the first round of experiments may indicate that the user's inability to articulate literal entry (unmodified by the decoder) may be negatively impacting entry rates. This possible effect was reframed as a design principle and further investigated.

Stage 6. Validate Refined Application

Finally, the effectiveness of the system and the design process is validated by evaluating its performance in a realistic application setting or more general test case. In Chapter 6, the developed text entry system is deployed and evaluated in a free roaming spatial annotation

task. In Chapter 7, the interface refinement strategy is deployed on a fundamentally new and unfamiliar design problem to highlight its generality.

3.3 Motivation for Choice of Methodology

The motivation behind the described research methodology stems from two key factors: i) limited broad user exposure to mixed reality; and ii) limited established design heuristics or frameworks for mixed reality. At present, there is limited broad exposure to mixed reality interfaces among the general population. Given this lack of an established user base or dominant application set it is difficult to conduct more established methods of investigation based on observation and/or survey. Similarly, without established design heuristics and frameworks specifically targeting MR interfaces, there is comparatively limited applicability of analytical approaches to the design problems highlighted. In practice, this eliminates large segments of the range of research strategies available over the unobtrusive-obtrusive and universal-particular spectrum [117].

Nevertheless, a major benefit of the outlined approach is that the derived principles feed into generalisable design guidance. In contrast to point studies of novel mixed reality features, this approach provides insight into how particular design decisions may influence user performance and experience. This promotes the establishment of recognised design processes as more principles are identified and tested.

3.3.1 Limitations

The focus on identifying high level design principles can reasonably be criticised for ignoring the broader challenges associated with standard application design. For example, this thesis makes no consideration for user groups with particular needs. The justification for this narrowness in focus is the fact that the likely initial users of the systems examined in this thesis are early adopters. As the underlying technology advances and larger numbers of users engage with it, however, it will be necessary to expand the methodology to encompass the broader design considerations.

3.4 Scope

The scope of this investigation is constrained in two key ways. First, although there are clearly advantages to treating other more conventional interaction contexts probabilistically, this project focuses exclusively on mixed reality. As highlighted earlier, designing for mixed

reality is particularly challenging for several key reasons. This difficult context more clearly distinguishes the advantages of the proposed methods. Furthermore, this area has received remarkably little attention from the HCI community due to recent technological advances.

Second, the lower-level systems that support and enable the probabilistic user interface are excluded from investigation. This scoping relates to how the system boundary is drawn to constrain which sources of input are to be treated as uncertain by the application interface. For example, the hand tracking or localisation sub-systems of modern augmented reality HMDs may be sensibly incorporated within the boundary of the user interface and treated in a probabilistic fashion to infer user actions and behaviour. In this project, however, the boundary of the user interface is constrained to the software layer exposed to typical application developers. This helps to promote hardware and device agnosticism and therefore, wider applicability of the specific solutions identified.

In part, this limitation of scope is also enforced by the fact that current HMD hardware does not typically report the uncertainty in its sub-systems. As the technology matures and the benefits of a probabilistic treatment are more widely recognised, device manufactures may choose to address this limitation. Ultimately, the choice on where to draw the system boundary is likely also to be influenced by the magnitude of uncertainty in different input signals. For example, understanding and separately modelling the particular noise characteristics of a low quality tracking system may support a better probabilistic treatment than simply subsuming the tracking noise into the input characteristics of user actions. Considering these various factors, others may choose to apply different system boundaries but the scoping applied in this thesis is sufficient to highlight the performance potential and flexibility of probabilistic user interface design.

3.5 Choice of Case Studies

As highlighted in Section 1.3.2, two secondary research objectives in this thesis are to: i) demonstrate the effectiveness of a probabilistic treatment in a range of applications; and ii) demonstrate a variety of effective probabilistic approaches. The case studies of Chapters 4 and 6 both examine the problem of supporting effective text entry in virtual and augmented reality. This problem domain is both highly relevant (text entry represents core foundational functionality for next-generation MR HMDs) and appropriate for a probabilistic treatment (language and typing exhibit high levels of predictability). Furthermore, text entry using conventional computers or smartphones is a familiar task to most users. Exposing users to this familiar task but within a fundamentally new and different interaction environment provides insight into the confounding effects produced by MR HMDs.

While Chapters 4 and 6 focus on the generation of textual content, Chapters 5 and 7 examine the principles relevant to its perception and presentation. The case study described in Chapter 7 examines the more general task of designing a user interface in mixed reality. This addresses the challenge identified in Chapter 1 of high dimensionality in, and general unfamiliarity with the design space. The interfaces examined in Chapter 7 specifically examine several design parameters relevant to the presentation of textual content. The application of crowdsourcing to facilitate rapid iteration through user testing serves to highlight how a probabilistic treatment can be both generalisable and scalable. Bayesian optimisation is the technique applied in this case study, as distinct from the map decoder based inference of Chapters 4 and 6.

The case study presented in Chapter 5 also leverages crowdsourcing to demonstrate how probabilistic models describing user populations can be leveraged to adapt the MR interface according to context. This case study also addresses the challenge identified in Chapter 1 related to uncertain deployment contexts. The concrete design problem examined is how the appearance of textual content can be dynamically modified to accommodate the physical context that provides its background.

In overview then, the four case studies deliver good coverage across a range of application design challenges and demonstrate a suite of complementary techniques. They are thematically linked by the required core function in mixed reality to support the generation and consumption of textual content. Together they highlight the generalisability, scalability and effectiveness of probabilistic user interface design.

3.6 Summary

The comparatively unexplored design space around interactive mixed reality applications necessitates a specialised research methodology. The methodology outlined in this chapter reflects an attempt to bring a structured and repeatable approach to the challenge of probabilistic user interface design for mixed reality. Clearly, alternative methodologies are available but the approach taken enables this research to not only tease out how the probabilistic nature of MR interfaces can be exploited but also how particular design decisions influence user performance and experience. This serves to contribute to the foundation of established principles and design guidelines for mixed reality interfaces.

Chapter 4

Characterisation

This chapter seeks to answer *Research Question 1*: *How can a designer obtain an understanding of the probabilistic characteristics of an interface; and, how can this understanding inform design in mixed reality?* The two parts of this question are investigated through a case study examining a fundamental human-computer interaction task: text entry. Given the exciting perceptual and interactive opportunities offered by mixed reality, it may seem unusual on first consideration to focus on the comparatively mundane task of text entry. To ensure these paradigms are more broadly usable and effective, however, it is necessary to also deliver many of the conventional functions of a smartphone or personal computer. It remains unclear how conventional input tasks, such as text entry, can best be translated into mixed reality.

This chapter describes a detailed characterisation of the performance potential of four alternative text entry strategies in virtual reality. These four strategies are selected to provide full coverage of two fundamental design dimensions: i) physical surface association; and ii) number of engaged fingers. Specifically, this chapter describes an evaluation of typing with index fingers on a surface and in mid-air and typing using all ten fingers on a surface and in mid-air. The central objective is to evaluate the human performance potential of these four typing strategies without being constrained by current tracking and statistical text decoding limitations. To this end, an auto-correction simulator that uses knowledge of the stimulus to emulate statistical text decoding is introduced. Additionally, high-precision motion tracking hardware to visualise and detect fingertip interactions is utilised. The characterisation shows that alignment of the virtual keyboard with a physical surface delivers significantly faster entry rates over a mid-air keyboard. Also, users overwhelmingly fail to effectively engage all ten fingers in mid-air typing, resulting in slower entry rates and higher error rates compared to just using two index fingers.

In addition to identifying the envelopes of human performance for the four strategies investigated, Section 4.5.2 provides a detailed analysis of the underlying features that distinguish

each strategy in terms of its performance and behaviour. The implications of these results for the design of a fully-fledged text entry system for mixed reality is then discussed in Section 4.6.

4.1 Introduction

Text entry is a fundamental human-computer interaction task [18]. Even in novel interaction environments, such as those enabled by virtual and augmented reality, text entry is an essential feature for synchronous and asynchronous communication, annotation and documentation. The delivery of seamlessly integrated and efficient text entry methods can potentially improve engagement and sense of presence by avoiding the need to switch between input devices or platforms. However, how to best deliver a productive and enjoyable method for entering text in such environments remains an open research question.

Recent advances in speech recognition have increased the popularity of voice transcription as a text entry method. Speech entry rates are fast and recent technical advances mean that accuracy rates are also comparable with conventional text entry methods [154]. However, privacy considerations and ambient noise mean that speech-to-text is not always viable. In reality, voice and touch-based text entry are complementary. Ultimately, a robust text entry solution for mixed reality will likely be delivered through a range of different and complementary input methods. Delivering a touch-based text input method that is familiar to users and leverages existing typing skills is therefore a desirable feature in mixed reality.

This chapter presents an exploratory study examining the human performance envelopes, that is, the feasible range of text entry rates and error rates, of four alternative touch-based typing configurations in VR. Results are reported from a controlled experiment with 24 participants that examines two fundamental design parameters: 1) aligning the keyboard with a physical surface compared to having the keyboard float in mid-air; and 2) typing with all ten fingers compared to just the two index fingers. This investigation thus concentrates on two fundamental factors likely to reflect the different circumstances of use of a virtual keyboard.

The central objective is to characterise the probabilistic qualities of the text entry interface in order to understand the empirical human performance potential of particular text entry strategies, independent of current device and software limitations. This motivates the elimination of tracking and statistical text decoding performance as factors in the experiment, as current state-of-the-art tracking and statistical text decoding performance would effectively result in an artificial ceiling effect on text entry rates. To address this concern, the VR typing setup developed for this investigation uses precision finger tracking provided by an OptiTrack motion capture system and robust auto-corrections delivered through a simulated statistical text decoding strategy (based on relaxed point-based matching [90]). The focus on VR over AR

is also motivated by efforts to control for confounding variables in the experiment. However, many of the investigated principles are anticipated to be directly relatable across target display environments (see Section 4.7.1 for further discussion).

In addition to investigating potential entry and error rates, the recording of precision fingertip tracking data facilitates the examination of more subtle micro metrics of performance and behaviour. These micro metrics include: touch accuracy variation over the layout; variation of mistypes over the layout; press depth, duration and velocity; as well as hand and finger usage proportions. These micro metrics assist in refining an understanding of touch-based typing requirements in two important ways. First, understanding the behaviour of the fastest typists helps formulate reasonable minimum requirements for tracking fidelity. Second, understanding what behaviours yield high entry rates and low error rates can inform the design of the layout and interactions in order to guide users towards more optimal typing behaviour.

The primary contributions of this chapter are:

1. A quantitative evaluation of the performance potential of four feasible touch-based keyboard text input strategies for VR covering two key design dimensions.
2. A probabilistic characterisation delivering a provisional set of indicative micro metrics of performance and behaviour that inform the design of a fully functional keyboard.

In highlighting the above contributions, this chapter begins by first reviewing the related work on touch-based typing in virtual and augmented reality. The system and apparatus used in the controlled experiment is then described along with details of the experimental protocol. The key results of the experiment are highlighted and then qualified and discussed. Finally, the conclusion of this chapter summarises the main results and revisits the contributions in the context of the broader objective of delivering a productive and enjoyable text entry system tailored to mixed reality.

4.2 Related Work

This section examines the literature relevant to enabling productive text input in VR. The research in this area is particularly interesting given the very broad range of strategies explored. Early work in this area experimented with handwritten notes (e.g. Poupyrev et al. [146]) and audio annotations (e.g. Harmon et al. [61], Verlinden et al. [181]). The potential of glove-sensed hand gestures (e.g. Rosenberg and Slater [153], Kuester et al. [93]) has also been widely explored. Bowman et al. [14] investigated the relative merits of these and other approaches by examining speech, glove, pen and chording keyboard approaches in a single experiment: entry rate results were 13 wpm, 6 wpm, 10 wpm and 4 wpm respectively. Speech-to-text has

advanced significantly over the past decade and now provides a viable and widely implemented input strategy for head-mounted displays (HMDs). For this reason, no further focus is given in this review to speech-based text entry research.

To help compartmentalise these various approaches and contextualise their relative advantages and disadvantages, the following categorisation is applied: virtual Qwerty keyboards; non-Qwerty layouts; and input device/glove based approaches. Also relevant to this study is work which examines the more rudimentary behaviours of how people type and these are examined at the end of this section.

The familiarity of the standard Qwerty layout strongly motivates its use in mixed reality settings. The significant challenge becomes how to effectively capture input on that layout. ARKB [98] describes an early implementation for vision based tracking of fingertips enabling multi-finger typing in AR. Tracking accuracy and latency were noted to be major challenges to usability given the technology limitations at the time. Leveraging significant technology advancements, ATK [198] makes use of the Leap Motion to demonstrate a full 10 finger mid-air keyboard supported by a probabilistic decoder. Participants achieved 29 wpm after one hour of practice although stimulus phrases were selected to ensure only words in the known vocabulary were included. VISAR [38] also leverages probabilistic decoding, in an approach derived from Vertanen et al. [187], and demonstrates single-finger mid-air text input specifically tailored for AR HMDs. After various refinements, including the provision of error-tolerant word predictions, the touch-based approach yielded a mean entry rate of 17.8 wpm. Although focussing on interaction with large wall displays, Markussen et al. [114, 115] examine both discrete and gesture-based approaches for mid-air text entry.

The challenges of delivering robust touch-based interaction with a virtual keyboard has also motivated the investigation of alternative articulation strategies. Yu et al. [199] compare tap selection on a gamepad, gaze-dwell and gaze-gesture articulation strategies for typing in VR: entry rates achieved were 10.6, 15.6, and 19.0 wpm respectively. With further refinement of the gaze-gesture approach, participants were able to reach an average entry rate of 24.7 wpm when typing the same 10 phrases repeatedly.

Non-Qwerty layouts have received attention as a way to mitigate restricted input and/or visual space on mixed reality HMDs. For example, Palmtype [190] re-appropriates the palm as a display and interaction surface for a virtual keyboard in AR. This approach builds on the more general body of research demonstrating the benefits of passive haptic feedback for interactions in virtual environments [101, 85]. Both Grossman et al. [55] and Yu et al. [200] examine simplified input strategies that accommodate the limited interaction surface on smart glasses. Other exotic layouts and interaction methods include: a 12 key keyboard with selections made by a combination of taps and slide gestures [134]; and a radial layout rotated using a controller

[201]. Eliminating the need for a layout altogether, AirStroke [128] allows users to input characters in mid-air using the Graffiti alphabet. Such approaches are, however, clearly rate limited, but AirStroke [128] applies a clever strategy of allowing the non-gesturing hand to select word predictions.

Finally, hand held input devices (e.g. Twiddler [110]) and gloves (e.g. Rosenberg and Slater [153], Kuester et al. [93]) offer a potential avenue for delivering text input functionality in AR and VR. Several of the studies previously mentioned use game controllers as an alternative means for articulation. While such approaches may be appropriate in certain circumstances and applications, a significant downside is that they encumber the user. Further, users' existing typing skills and keyboard layout awareness are not easily leveraged in these approaches.

More general efforts to better understand and exploit typing performance and behaviour in novel input arrangements also inform this study. Findlater and Wobbrock [42] examine the potential for adaptive keyboard layouts in 10 finger touchscreen typing that update based on observed patterns of behaviour. Influence is also taken from Sridhar et al. [171] who take a considered approach to understanding dexterity as a precursor to building a mid-air finger articulation based text input system.

In summary, the literature offers a somewhat confusing landscape of different strategies for supporting text entry in mixed reality. It can be difficult to understand the raw potential of these various approaches given the different experimental protocol choices and technical limitations that inevitably colour these results. This factor is, in part, what motivates the examination of high-level design choices using an experimental protocol that is inherently optimistic in determining envelopes of human performance but robustly supports relative comparison between the techniques under investigation within the same experiment. There are clearly many factors which ultimately determine the entry rate potential of a particular input strategy in practical use. Rather than pursuing and demonstrating a 'practical' text entry system at this juncture, this chapter instead takes an exploratory approach to characterising the text input task that will inform subsequent design efforts.

4.3 Approach

This study has three key objectives. These are:

1. Determine the human performance potential of alternative text entry strategies for MR.
2. Capture hand and finger tracking data representative of typical typing behaviour.
3. Mine the recorded tracking data to identify implications for the design and development of a fully functional keyboard and input system tailored to these strategies.

Objectives 1 and 2 above are pursued in parallel. To facilitate the examination of performance 'potential' and to ensure user typing behaviour is representative of a properly functioning virtual keyboard, an express decision was made to test an 'ideal' system where conventional tracking and statistical text decoder limitations are removed. Therefore, precise marker-based tracking (OptiTrack) is employed and a simulated auto-correction strategy is introduced.

Clearly the elected approach yields an optimistic assessment given that currently available low-cost head-mounted or remote finger tracking technology cannot achieve the accuracy levels of an OptiTrack setup. Furthermore, the effectiveness of the simulated auto-corrections may exceed the performance of a conventional statistical text decoder naïvely applied. Nevertheless, the approach does effectively inform the development of next-generation text entry methods for mixed reality by: i) determining which strategies are worthy of practical examination under conventional device limitations; and ii) highlighting technical requirements for tracking and statistical text decoder components to enable high levels of typing performance. The pursuit of objective 3 above informs an understanding of this second point. This analytical approach is inspired in part by prior work performed by Feit et al. [41] and Dhakal et al. [30].

4.4 Test Bed for High Performance Text Entry in VR

In preparation for the controlled user experiment, a test bed was developed for examining text entry strategies delivering high-precision finger tracking and the illusion of robust auto-corrections. These two main system components, in addition to the virtual environment in which they are embedded, are described in detail in the following sections.

4.4.1 Finger Tracking

Precision (sub-millimetre) fingertip tracking is provided by an OptiTrack motion capture system (using Prime 13 cameras). A rigid markerset is attached to the back of each hand to provide position and orientation tracking. Individual markers are then temporarily attached to participant fingertips (on the fingernail). The HMD is also tracked using a separate rigid-body markerset. The experimental setup is shown in Figure 4.1. The position and orientation of each palm is coarsely represented by the purple rectangular prisms shown in Figure 4.2. The fingertip positions are represented by purple spheres.

4.4.2 Simulated Auto-Correction

The behaviour of a standard error correcting statistical text decoder is approximately replicated by performing point-based matching. This approach is introduced by Kristensson and Zhai [90].

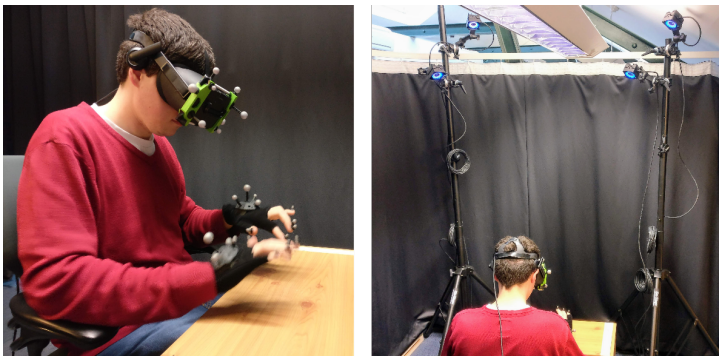


Fig. 4.1 User shown typing in mid-air with HMD, hand and fingertip tracking markers (left). OptiTrack camera setup for precision marker tracking (right).

Fig. 4.2 The keyboard, hands (represented by the purple prisms and spheres) and virtual work environment as viewed in the VR headset.

The point-based matching procedure determines the number of substitutions, insertions or deletions required to align the observation sequence with the target sequence. Importantly, however, it is possible to apply a tolerance on what is considered a successful match. Through pilot studies a suitable tolerance of 2.5× the nominal key radius was identified. Several example traces illustrating this approach are presented in Figure 4.3.

It is important to note that this approach only works because participants must type known preset stimulus phrases. To mimic the behaviour of an auto-correcting decoder, the known words in the stimulus phrase are supplied to the simulated auto-correction component. The latest observation points are sent to the decoder upon particular input events, e.g. space and punctuation (other than apostrophe). If at least 80% of the observation sequence matches the target sequence for a given word in the stimulus phrase, it will be substituted as an auto-correction. Clearly this penalises shorter words, however, such is also the behaviour of a standard statistical text decoder given limited observation points. Once a word in the

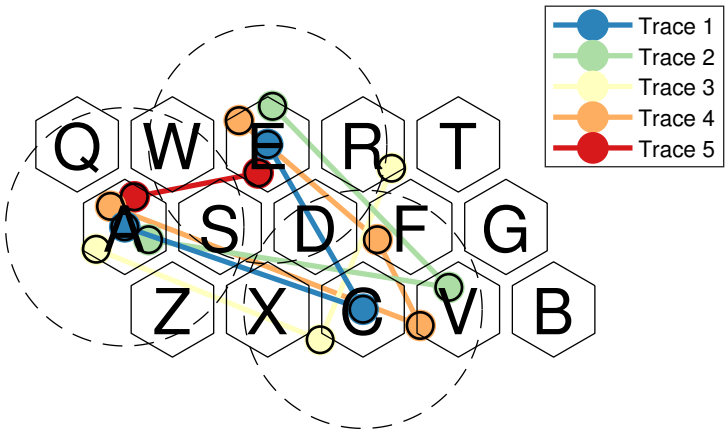


Fig. 4.3 Five example observation sequences (traces) for typing the word 'ACE'. Trace 1 is the target sequence (ideal observations) where the centre of every key is hit. Trace 2 is a good observation sequence in that all observations are within the tolerance of the targets, even though 'V' is actually struck instead of 'C'. Trace 3 is a bad observation sequence since the last observation is outside the tolerance for the target 'E'; this is a substitution error (substitution edit required). Trace 4 is a bad observation sequence since four points are observed; this is an insertion error (deletion edit required). Trace 5 is a bad observation sequence since only two points are observed; this is an omission error (insertion edit required).

current stimulus phrase is substituted, it is removed from the list used to evaluate subsequent observation sequences.

4.4.3 Virtual Environment and Keyboard

A virtual work environment was constructed to provide a thematically relevant context for the text entry task. This environment featured a simple wooden desk against a painted wall. The virtual keyboard and work desk are visible in Figure 4.2. The surface of the virtual table was aligned with the surface of a physical table in the experiment space. The table can be seen in Figure 4.1.

A full Qwerty virtual keyboard was designed with all 26 characters and a reduced set of punctuation (,'?!.). Keys are placed with compact tessellation, with each key having an apparent diameter and separation of approximately 25 mm. The top row of keys (*Q*–*P*) therefore has an apparent width of 250 mm making it roughly 30% wider than the top row of a typical physical keyboard (190 mm). The two-dimensional keyboard layout is illustrated in Figure 4.4.

Keyboard touch events are generated when a spherical collider attached at the fingertip marker location first intersects with the keyboard detection plane. The collider attached at each fingertip marker site has a fixed size since no online association of markers to specific

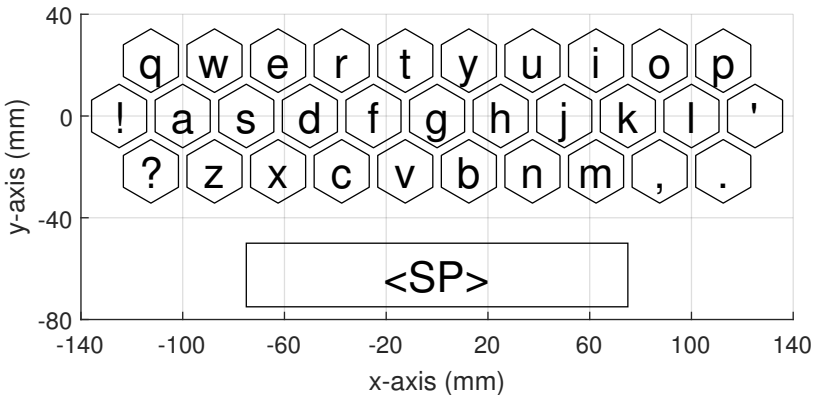


Fig. 4.4 The keyboard layout used in the experiment. Note the reduced set of punctuation.

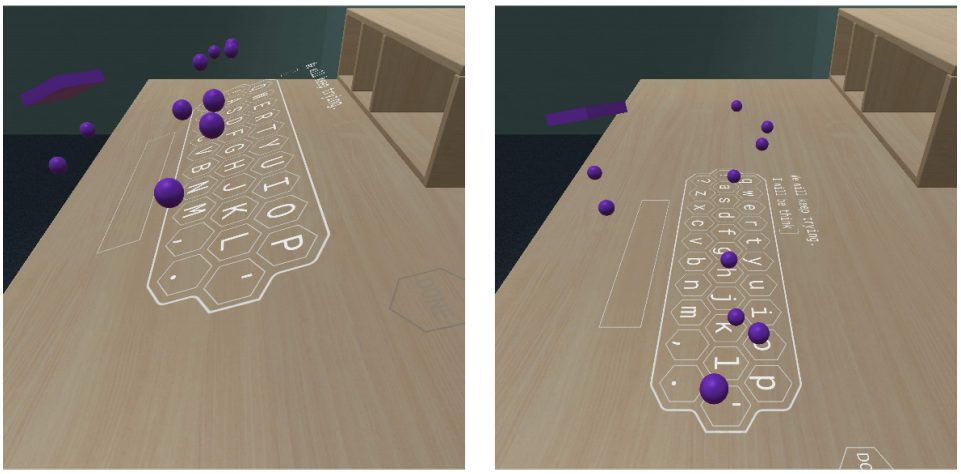


Fig. 4.5 The keyboard shown in its mid-air configuration (left) and aligned with the table (right).

fingertips is performed. Note that to generate subsequent touches with the same fingertip, the collider must completely leave and re-enter the detection plane. A simple visual animation at the touch point, synchronised with an audible click sound, provides feedback indicating a generated touch event.

The stimulus phrase is shown in the top row above the keyboard. Entered text is shown immediately below this. For the purpose of the experimental task, a *DONE* key is included for users to press when their entry is complete. The interface layout experienced by participants can be seen in Figure 4.2. Figure 4.5 illustrates the positioning of the keyboard in the mid-air and surface-aligned conditions.

Importantly there is no backspace or delete key. As described later, participants were given no opportunity to correct errors.

4.5 Experiment: Typing Performance Potential

The experiment required participants to complete a text transcription task. This task was performed in the following four typing conditions:

- MA2: Mid-air, two (index) fingers only
- SUR2: Aligned with physical surface, two (index) fingers only
- MA10: Mid-air, all fingers
- SUR10: Aligned with physical surface, all fingers

After obtaining ethics approval for the study, a call for participants was placed on a public-facing university website and 24 people (10 female, 14 male, median age = 25) from a range of disciplines and professions were recruited. The condition order was fully balanced to address potential learning effects (i.e. no two participants experienced the same order of conditions). The experiment was split into two sessions, with each session examining two of the four conditions. Participants were required to perform these sessions on separate days but with no more than two days break between sessions. A single session would typically run for between 1.5 and 2 hours, resulting in a total experiment time of between 3 and 4 hours. As part of the participant briefing, participants were instructed to notify the researcher if they experienced any VR induced nausea so that the experiment could be suspended. Note that this situation did not arise.

The experiment controlled for posture by enforcing a seated position. In addition, participants were not permitted to rest any part of their hand or arm on the table in the mid-air conditions, but were free to do so in the surface aligned conditions.

Within each condition, participants were presented with 10 practice sentences and 160 test sentences. During the practice sentences, participants were encouraged to attempt different typing strategies and to develop an understanding of the keyboard behaviour.

The 160 test sentences were split into four blocks of 40 sentences with the opportunity for a short break between each. Stimulus sentences were taken from the extended Enron mobile message dataset [185] and filtered based on phrases containing four words or more, and 40 characters or less. Stimulus phrases were selected from this subset without replacement, such that participants never saw the same sentence twice. In summary, a total of $(24p \times 4c \times 160s)$ 15,360 test entries were captured in this experiment.

To remove error correction time as a confounding factor in the experiment, no backspace or deletion functionality was provided by the keyboard. Participants were instructed to type as accurately as possible, but in the event of an error, to continue typing as if no mistake had been made.

4.5.1 Results

The results of the controlled experiment are summarised in this section. First, the human performance potential of the four conditions in terms of entry and error rates are examined. Later, the various micro metrics that yield a greater understanding of underlying factors that explain user performance and behaviour are explored. Finally, the participants' qualitative feedback and general observations of typing behaviour in VR are reviewed.

Performance Potential

The key metrics describing performance in text entry are entry and error rate. The standard metric for entry rate is words per minute (wpm), that is, number of words entered divided by time taken. In practice, the numerator is an effective word count where a nominal word length of five 'keystrokes' is used (including spaces). Therefore, the effective word count is the entered phrase length minus one (since timing starts from the first touch) divided by five. To highlight the distinction between the standard assessment of entry rate and the investigation incorporating simulated auto-corrections, the measure, wpm_{sim}, is used.

Error rate is typically reported as Character Error Rate (CER), which is the minimum number of character insertion, deletion and substitution operations that transform the response text into the stimulus text, divided by the length of the stimulus text. However, given the behaviour of the simulated auto-corrections it is more appropriate to report error rates in terms of their geometric trace match. Therefore, the relaxed geometric match error rate is reported as ER_{relax}. ER_{relax} reflects the number of required edits normalised by the length of the observation sequence. The numerator is the count of substitutions, insertions or deletions required to align the observation sequence with the target sequence given a tolerance of $2.5\times$ the nominal key radius on each target key (this is consistent with the simulated auto-correction procedure outlined in Section 4.4.2). The denominator is simply the length of the observation sequence.

The entry and error rate results for all captured entries are summarised in Figure 4.6. Entry rates are observably higher in the on-surface conditions (SUR2, mean = 55.6 wpm_{sim} and SUR10, mean = 51.6 wpm_{sim}) than in the mid-air conditions (MA2, mean = 42.1 wpm_{sim} and MA10, mean = 34.5 wpm_{sim}). Using a repeated measures analysis of variance finds a significant effect for the keyboard test condition on entry rate ($F_{3,69} = 29.370$, $\eta_p^2 = 0.561$, $p < 0.05$). Using an initial significance level of $\alpha = 0.05$ and performing multiple comparisons with a Bonferroni correction (note that all subsequent reported multiple comparisons use this same procedure) shows a significant difference between all conditions except for between SUR2 and SUR10. This result suggests that physical surface alignment is an important factor in

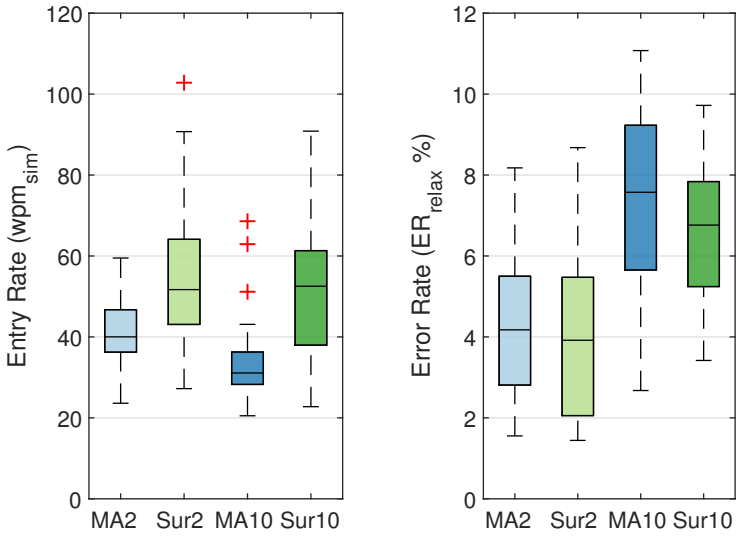


Fig. 4.6 Boxplots of participant mean entry rate (left) and relaxed error rate (right). In this and subsequent boxplots, red crosses indicate outliers based on $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$.

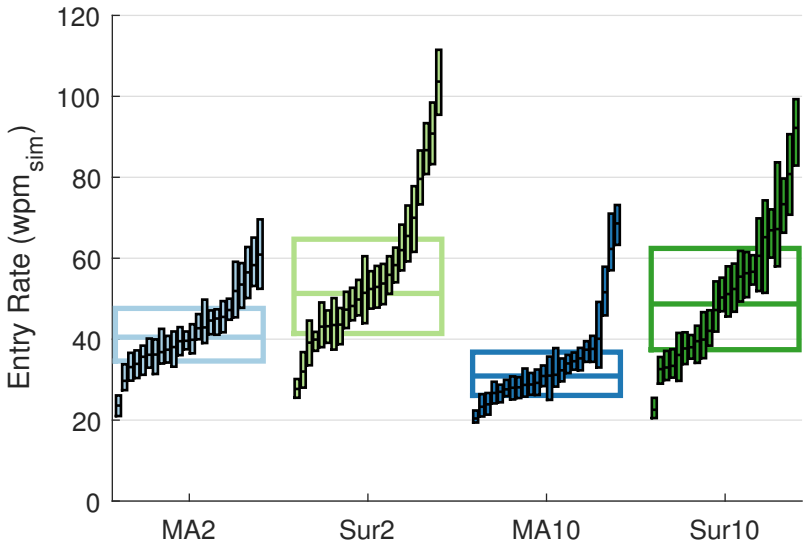


Fig. 4.7 Plot shows individual participant Q_1 (first quartile), median and Q_3 (third quartile) entry rates sorted by median entry rate to help better illustrate the structure of the distribution. Note that the plot only includes entries where the error rate is below 10%. Aggregate Q_1, median and Q_3 across all participants for an individual condition are also shown as the outer bars (coloured lines with white infill).

producing high entry rates. Section 4.5.2 will later examine the lower-level features that relate the presence of a physical surface to typing performance.

Interestingly and somewhat counter-intuitively, the ten finger conditions (MA10 and SUR10) do not yield significantly faster entry rates than their two finger alternatives. In fact, having ten fingers in mid-air appears to be detrimental to performance. This result correlates with the significantly higher error rates in the ten finger conditions ($F_{3,69} = 31.431$, $\eta_p^2 = 0.577$, $p < 0.05$). The effect is significant between the two and ten finger conditions but not within each. Although the experiment protocol did not enforce corrections, high error rates will typically lead to a negative impact on uncorrected entry rates: users pause to re-evaluate their place in the phrase and/or make more careful and precise movements to avoid further errors.

Figure 4.7 provides an alternative perspective on the entry rate results. Here the interquartile range is plotted for each participant. Only entries where the error rates were below 10% are included in this plot. Note that within each condition the plot order is sorted based on participant median to better illustrate the distribution. It is interesting to note the clear upper tail effect is more prevalent in certain conditions. This will be examined in more detail later in Section 4.5.4 when the metrics of the high and low performing participants are analysed.

4.5.2 Micro Metrics of Performance and Behaviour

This section examines a collection of lower-level features that are key determinants of typing entry and error rate. These features are subsequently referred to as micro metrics of performance and behaviour and they represent a finer grained characterisation of the text entry system and the user.

These features help reveal what aspects of the typing task are most influenced by the different conditions. For example, in the following section the accuracy of touches over the layout is examined. This analysis finds that higher accuracy is achieved in the two-finger conditions. Conversely, ten finger typing yields less accurate touches and this result correlates closely with heightened error-rates identified for conditions MA10 and SUR10.

Touch Accuracy

Figure 4.8 provides a summary representation of the touch accuracy variation over the layout. Note that these plots are generated from entries where the error rate was below 10% to ensure reasonable confidence in the realignment of the ideal and observed sequence. The ellipse on each key reflects the centroid and covariance of the touches associated with that key. Recall that the relaxed point-based matching used in delivering the simulated auto-correction

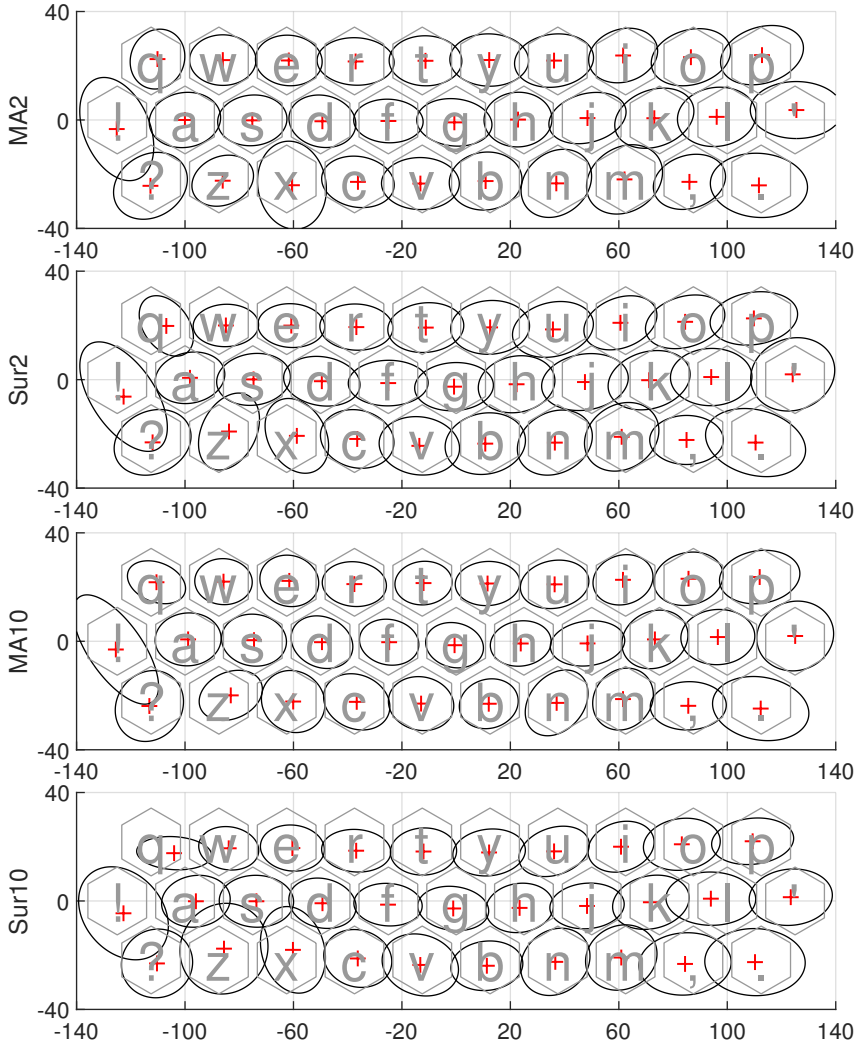


Fig. 4.8 Touch point covariance for each key over the layout represented as ellipses. From top to bottom: MA2, SUR2, MA10, and SUR10. Ellipses describe the 50% confidence interval. The variance in touch error is visibly higher in the *x*-axis than the *y*-axis across all conditions.

behaviour meant that users could touch outside the bounds of the target key and still experience a successful auto-correction provided it was within the distance threshold.

Several interesting observations can be made from Figure 4.8. First, in all conditions the variance in touch error tends to be higher in the *x*-axis than in the *y*-axis. This feature is suggestive of more precise finger articulation (i.e. to switch between key row) than wrist and/or forearm articulation (i.e. to move over the layout laterally). When all touches are collapsed together, the standard deviation in the *x*-direction is approximately double that in the *y*-direction. The larger spread in the *x*-direction is consistent with touch accuracy observations over the layout made by Azenkot and Zhai [6] (examining single finger and single/dual thumb typing on a smartphone) and Shi et al. [168] (examining 10 finger typing on an interactive tabletop). The results of Azenkot and Zhai, however, suggest considerably higher precision in key targeting when typing on a smartphone.

Second, Figure 4.8 highlights the fact that touches are more precise at the centre of the keyboard than at the edges. The additional variation in touch error towards the edges generally appears to radiate away from the very centre of the keyboard. One likely interpretation of this result is the fact that typical strategies in standard typing involve maintaining the fingers in an approximate 'home' position. It can be logically reasoned that moving fingertips away from their 'home' position at high velocity may be introducing this 'smearing' effect on touch error radiating outwards. Azenkot and Zhai [6] also hint at this mechanism producing variation over the layout in their examination of typing on a smartphone.

Touch Errors: Substitutions, Insertions, Omissions

In this section, the distribution of common typing errors over the keyboard layout is examined. Understanding any relationship between key position and/or typing configuration may inform alternative strategies for addressing such errors. Standard mistypes fall into three categories: substitutions—an incorrect key is pressed; insertions—an additional undesired key is pressed; and omissions—a desired key is not pressed.

Figure 4.9 illustrates the frequency of the three main mistype categories over the layout for the four conditions. A frequently observed mistype among participants in the ten finger conditions was the pinky finger inadvertently generating key presses at the extreme edges of the layout. This is observable in Figure 4.9 as a high proportion of insertions on keys *Q!?P'*. for conditions MA10 and SUR10.

Another common mistype observed, but less visible in Figure 4.9, are omissions on commonly doubled characters such as *T*, *L*, and *O*. This error stems from participants failing to raise their finger sufficiently high to exit and re-enter the detection plane. This particular issue is investigated in more detail later in Section 4.5.2.

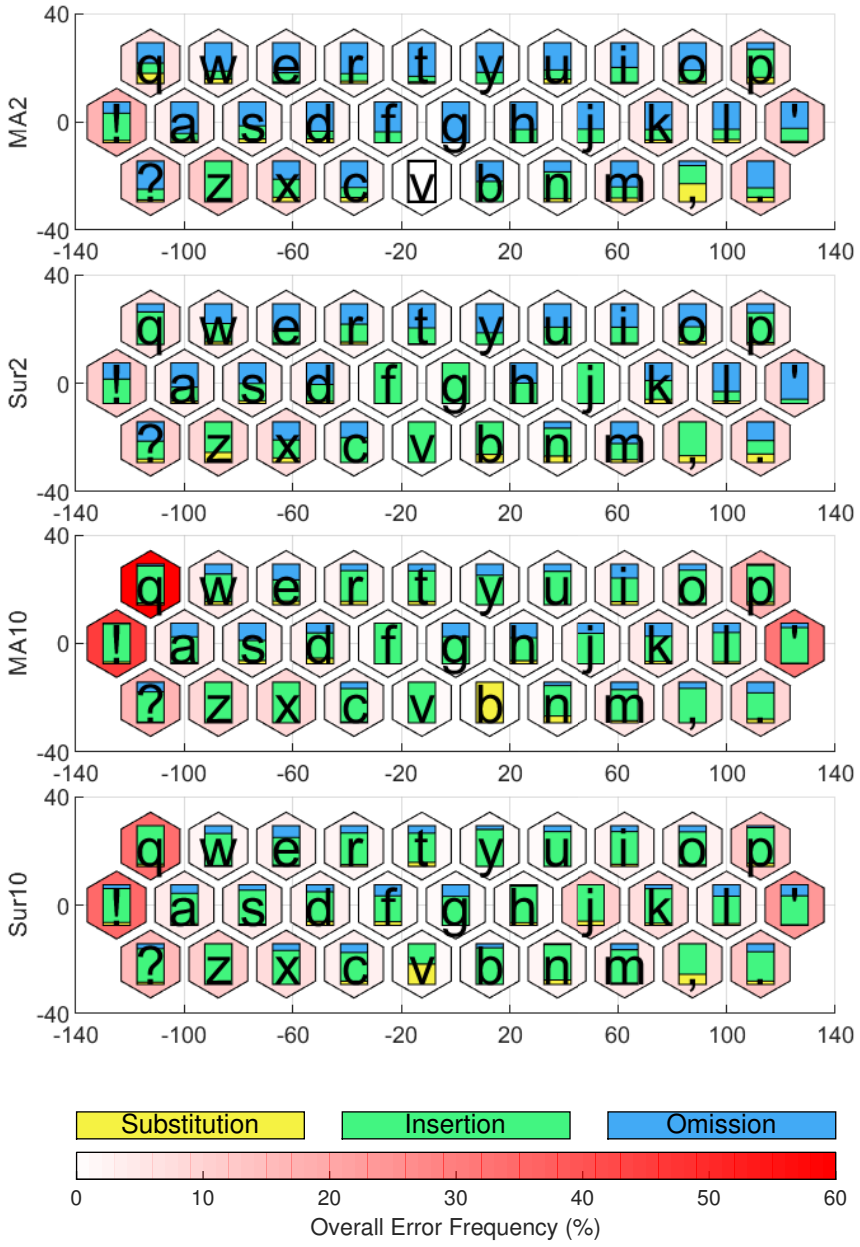


Fig. 4.9 Relative proportions of standard mistypes for each key over the layout. The stacked bar on each key represents the relative proportion of mistypes categorised into one of three groups: substitutions—an incorrect key press; insertions—an additional undesired key press; and omissions—a desired key is not pressed. The overall frequency of mistypes (of all categories) on a given key as a percentage of total presses for that key is represented by the red shading.

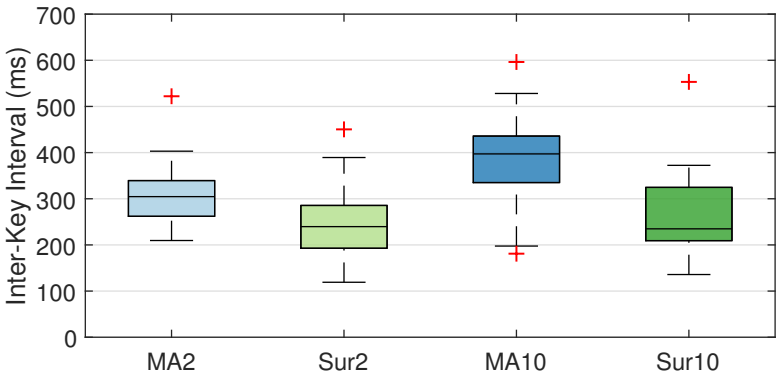


Fig. 4.10 Boxplots of participant mean inter-key interval. The inter-key intervals in the on-surface conditions, SUR2 and SUR10, were significantly faster than in the mid-air conditions, MA2 and MA10.

The most obvious distinction between the two finger conditions and the ten finger conditions is the dominant mistype being omissions for two fingers and insertions for ten fingers. This result is consistent with the higher error rates observed and general difficulty participants had in avoiding inadvertent touches with other fingers.

Inter-Key Interval (IKI)

The inter-key interval (IKI) metric reflects the time between key presses. It therefore correlates closely with entry rate. Figure 4.10 summarises the IKI in each of the four test conditions.

Repeated measures analysis of variance shows the test condition effect to be significant ($F_{3,69} = 48.318$, $\eta_p^2 = 0.678$, $p < 0.05$). The differences are significant between MA2 and all other conditions and MA10 and all other conditions. In other words, significantly faster IKIs were observed in the on-surface conditions (with interquartile ranges of approximately 200 to 300 ms). This is consistent with the faster entry rate results for these conditions. More time taken between key presses for MA2 (median of approximately 300 ms) and MA10 (median of approximately 400 ms) is correspondingly a significant contributor to the slower entry rates for these mid-air conditions.

It is likely that this additional time taken to transition between keys in the mid-air conditions is in part a result of the longer trajectory followed by the fingers. For on-surface typing the height of the fingertip is comparatively simple to regulate given the potential to rest the palm of the hand on the physical surface. By contrast, mid-air typing involves more challenging depth regulation given the lack of a fixed surface reference plane. The implication of this difference on press depth is examined later in this section.

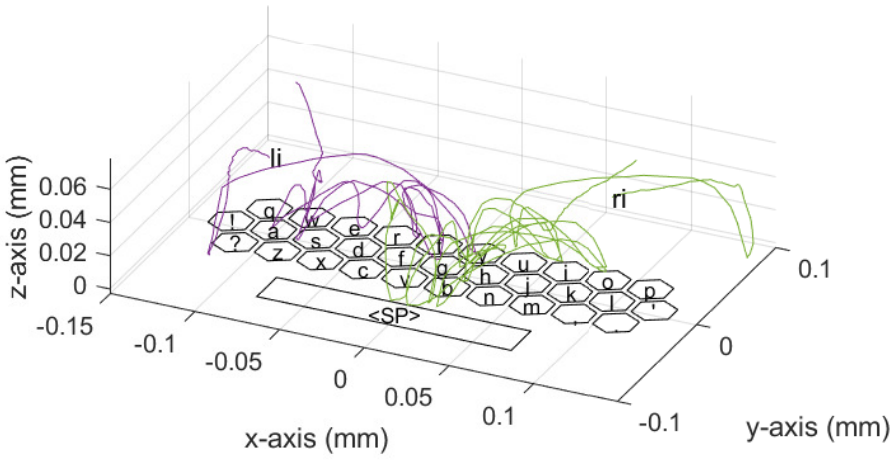


Fig. 4.11 An illustrative example of *P22* typing the phrase 'How are things with you?' with two index fingers in SUR2. Purple trace is left index finger, green is right.

Fingertip Trajectory

The captured tracking data enables the examination of lower-level features describing the fingertip trajectories in executing the typing task. Figure 4.11 provides an illustrative plot of the path traced by the tip of each index finger in *P22*'s execution of the phrase, 'How are things with you?' Figure 4.11 highlights the complex coordinated movement of fingers while typing. Figure 4.12 illustrates the z-component (in the direction out of the keyboard plane) of the same fingertip trace resolved into the keyboard frame.

A key objective of such analysis is identifying features that might help discriminate between re-positioning (i.e. preparing for a key press) and striking (i.e. executing a key press) motion of the finger. To this end, mean fingertip velocity as the touch event is first initiated is examined in all typing conditions. Figure 4.13 shows the velocity of each fingertip during the execution of the same trace shown in Figures 4.11 and 4.12. The press velocity is computed based on the three dimensional velocity as the fingertip enters the detection plane. With reference to the example shown in Figure 4.13, the press velocity is the velocity value at entry into each red shaded region since this region indicates the period during which the fingertip is inside the detection plane.

Figure 4.14 summarises the participant mean press velocity in each condition. A significant effect of test condition on fingertip velocity at touch time is observed ($F_{3,69} = 22.383$, $\eta_p^2 = 0.493$, $p < 0.05$). The differences are significant between MA2 and all other conditions and SUR2 and all other conditions. This result highlights the fact that the fingertip is travelling significantly faster when touches are generated in the two finger conditions than in the ten finger conditions. This result is intuitive when considering the fact that when only two fingers

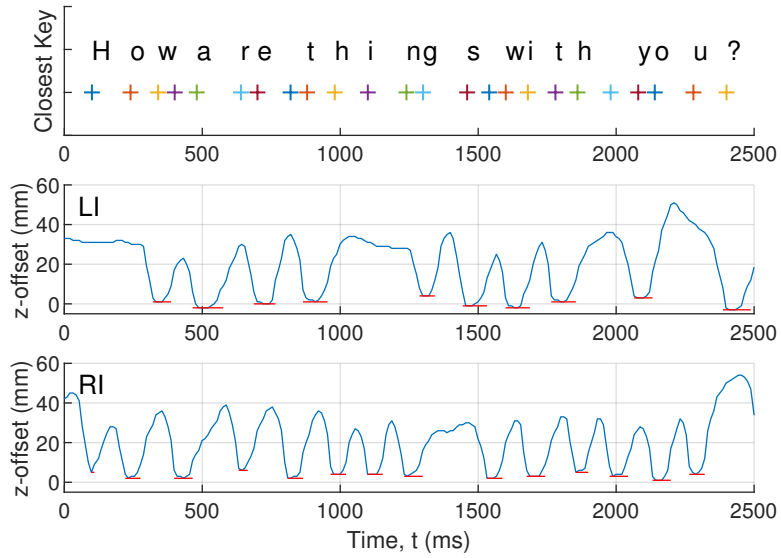


Fig. 4.12 The z-offset resolved into the keyboard frame for the trace shown in Figure 4.11. The depth, frequency and duration of touches can be easily observed in the z-offset trace generated by the left index (LI) finger (middle plot) and right index (RI) finger (bottom plot). The top plot shows the closest key at each touch event.

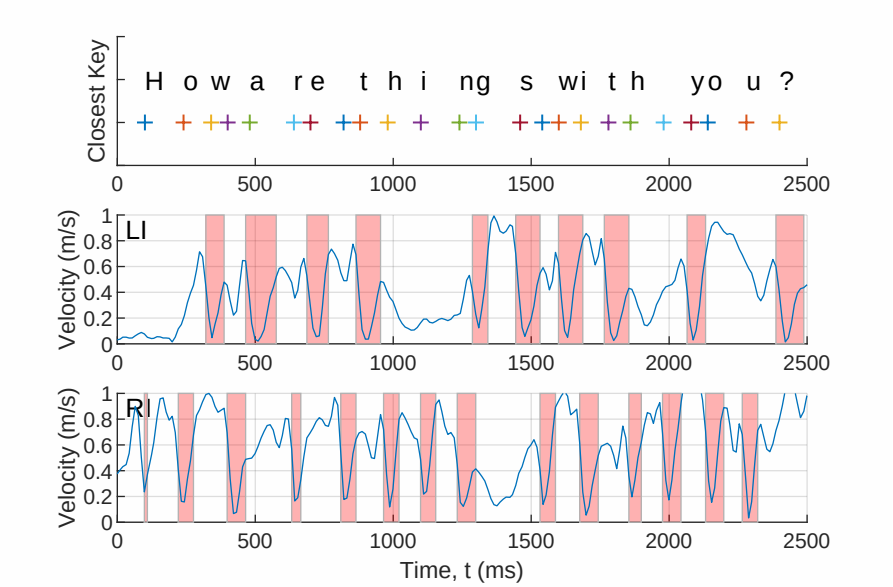


Fig. 4.13 Velocity of each fingertip during the example trace shown in Figures 4.11 and 4.12. Note that this is the three dimensional velocity and not just the velocity in the z-component. The red shading indicates the finger is inside the detection plane. LI: left index (middle), RI: right index (bottom).

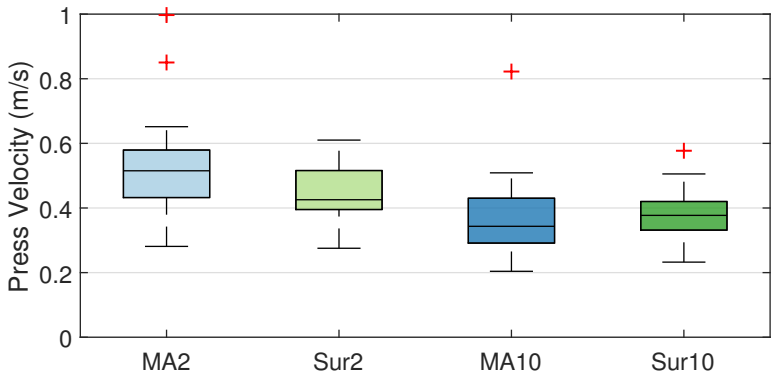


Fig. 4.14 Boxplots of participant mean press velocity. The press velocities in the MA2 condition are significantly faster than all other conditions.

are available, the motion between target keys must be faster to maintain a given entry rate. The significant difference between MA2 and SUR2 is likely a consequence of the absence of the physical limit and therefore no penalty (i.e. potentially painful striking of the surface with the fingertip) on high speed touches.

Press Duration

The press duration is the period of time spent inside the detection plane when executing a key press. This z-offset resolved in the keyboard frame (as shown in Figure 4.12) enables simple analysis of press duration. Figure 4.15 summarises the participant mean press durations for each of the test conditions. Shorter presses were observed in the surface-aligned conditions (SUR2, mean = 120.3 ms and SUR10, mean = 118.9 ms) than the mid-air conditions (MA2, mean = 142.9 ms and MA10, mean = 128.7 ms). The effect of the test condition is observed to be significant ($F_{3,69} = 9.017$, $\eta_p^2 = 0.282$, $p < 0.05$). Performing multiple comparisons, a significant difference is observed between MA2 and all other conditions. In other words, the presses in the MA2 condition last significantly longer than those in the two surface-aligned conditions as well as the ten finger mid-air condition. It is likely that the longer period spent within the detection plane is a consequence of deeper travel as examined in the following subsection.

Press Depth

The press depth is the maximum distance past the detection plane travelled by the finger. This measure is observable in Figure 4.12 as the local minimum in the z-offset at each of the touch events. The mean press depth in each condition is summarised in Figure 4.16. Clearly the

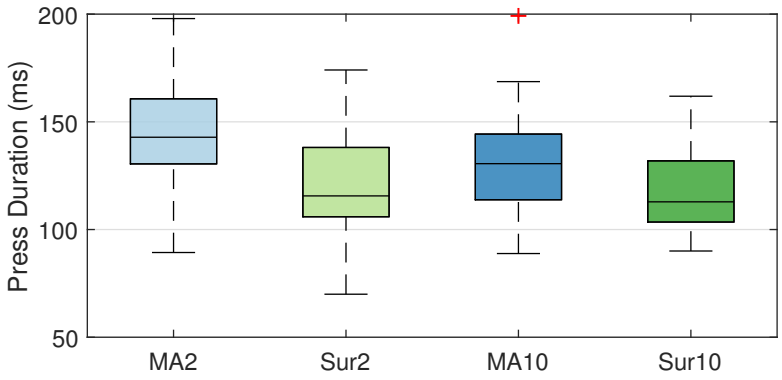


Fig. 4.15 Boxplots of participant mean press duration. The duration of presses in the MA2 condition is significantly longer than in all other conditions.

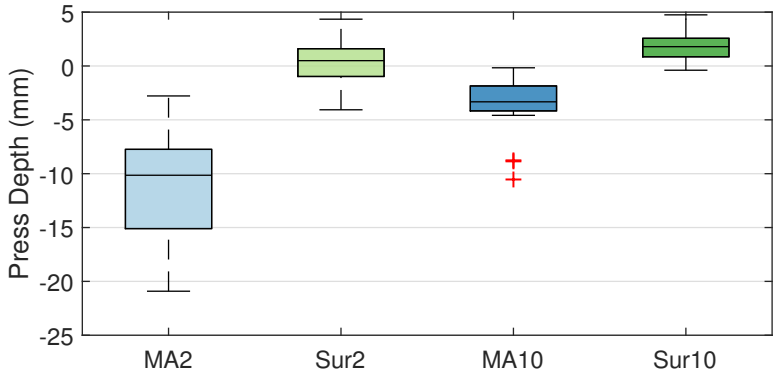


Fig. 4.16 Boxplots of participant mean press depth. The depth of press in both mid-air conditions is considerably larger than in the surface-aligned conditions as expected. The presses in the MA2 condition are significantly deeper than all other conditions.

press depth is physically constrained in the surface-aligned conditions. Recall, however, that touch events are raised based on a simple collision detection between a collider attached at the fingertip marker location and the keyboard plane. Since real time association of fingertips is not performed, the same fingertip collider size is used for all fingers. For this reason, as well as other potential sources of minor variation (e.g. marker attachment location, finger sizing, finger orientation while pressing), it is possible for touch events to occur before the physical limit is reached. As can be observed in Figure 4.16, these inadvertent touches are clearly more prevalent in the SUR10 condition.

A repeated measures analysis of variance shows the test condition to be a significant effect ($F_{3,69} = 99.461$, $\eta_p^2 = 0.812$, $p < 0.05$). The difference between all conditions is significant except for between the two surface-aligned conditions. For the mid-air conditions, the depth of touch is considerably larger in MA2 than MA10. One interpretation of this result is that when

only the index finger is engaged there is no penalty for deep movements into the detection plane. By contrast, when all fingers are engaged the user must be conscious of not moving other fingers into the detection plane. The deeper penetration into the detection plane for the mid-air conditions has a corresponding impact on press duration as highlighted in the previous subsection.

Press Reversal

As discussed in Section 4.5.2, a commonly observed mistype was an omission of repeated characters. This motivates examination of the trajectory followed by the finger in such circumstances. In particular, it is useful to determine what distance users will typically lift their fingers in order to indicate a 'press-and-release'. The press reversal therefore describes the minimum distance travelled to generate a double-tap of a repeated key. Figure 4.17 provides a helpful illustration of this motion. Here the user generates two presses on 't' in order to type 'little' and the press reversal here was 12 mm.

Figure 4.18 summarises the distribution of mean press reversal distances across the four test conditions. The effect of test condition is significant ($F_{3,69} = 63.887$, $\eta_p^2 = 0.735$, $p < 0.05$). Performing multiple comparisons, the difference is significant between all conditions except for between the two surface aligned conditions. The significantly shorter press reversals observed in the surface aligned conditions is likely a reflection of the higher degree of control that can be exercised when the palm of the hand is resting on a physical surface. Press reversal distances are highest in the MA2 condition which is consistent with the generally more pronounced movements observed in the mid-air conditions and also reflected in the analysis of press velocity and depth.

Hand and Finger Usage

The usage proportions for each hand and finger help give a sense of what typing behaviours are promoted by each of the typing conditions. Clearly, the two finger conditions constrain participants to type with index fingers only, yet understanding right/left dominance can be informative. More relevant, however, is the extent to which participants are able to fully exploit the full complement of ten fingers.

Recall that tracking was performed with passive markers and so there was no real time association of fingertips with markers. Nevertheless, such an association is relatively simple to apply in post-processing given the recorded left/right hand poses.

Figure 4.19 summarises the usage percentages across each of the test conditions for each touch event. Recall that the two finger conditions used only the index fingers so the hand usage

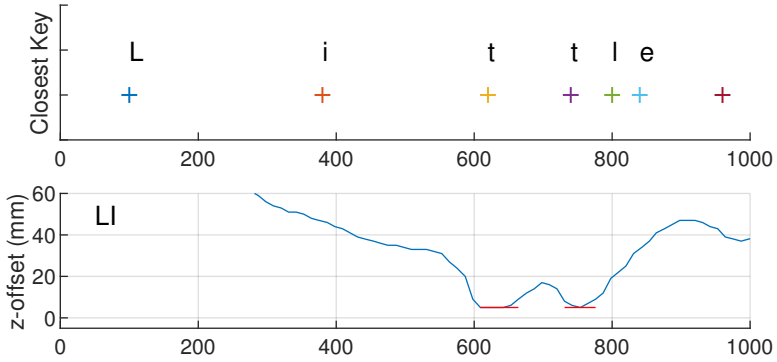


Fig. 4.17 Illustration of a double tap executed with the left index (LI) finger while typing the word 'little'. The top plot shows the closet key at each touch event while the bottom plot shows the z-offset of the left index finger as it performs a double tap on the 't' key.

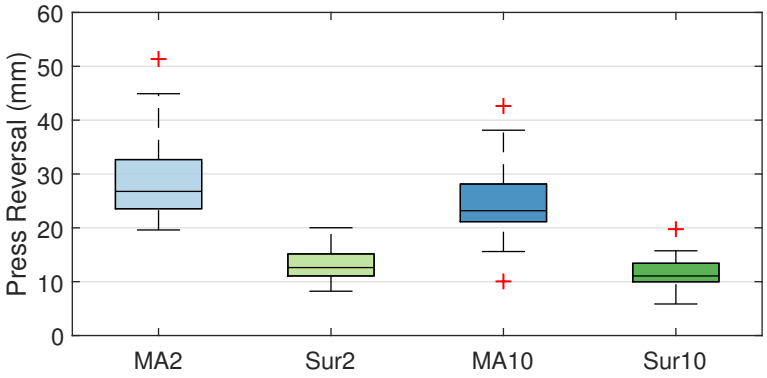


Fig. 4.18 Boxplots of participant mean press reversal. The reversal distance is significantly shorter in the surface aligned conditions.

percentage is the same as the finger usage. The index fingers are also dominant in the ten finger conditions, followed by the middle fingers then right thumb (used for space).

The usage percentage of the ring and pinky fingers is higher in MA10 than SUR10. Referring back to the common mistype results presented in Section 4.5.2, however, it is likely that this additional involvement of the outer fingers is actually a reflection of inadvertent insertions. Otherwise, the usage distribution in the ten finger conditions is remarkably similar.

4.5.3 Qualitative Feedback

After each experimental session, participants completed a short survey asking them to reflect on their experience with the typing conditions. Three statements examined the participant's perception of their speed ('How quickly were you able to type in this condition?'), accuracy ('How accurately were you able to type in this condition?') and comfort ('How comfortable

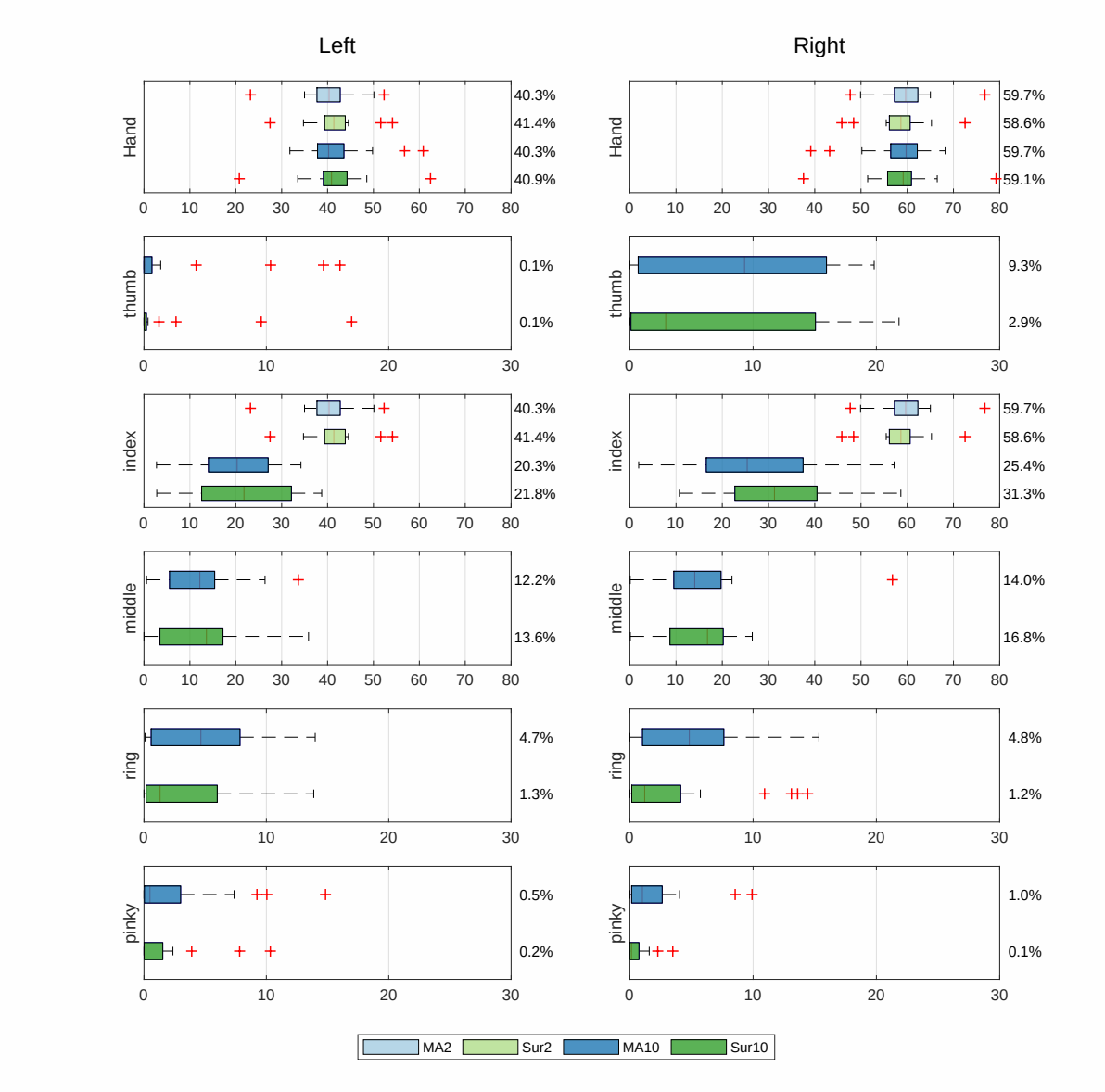


Fig. 4.19 Boxplots of participant mean hand and finger usage rate (%). Percentage values shown to the right of each plot are the median values. The MA2 and SUR2 conditions used only the index fingers so the hand usage percentage is the same as the finger usage. The index fingers are also the most frequently used in the MA10 and SUR10 conditions, followed by the middle fingers then the right thumb.

was typing in this condition?') while performing the typing task. Responses were collected on a five-point Likert scale (1: negative; 5: positive). The median response of the 24 participants to these statements are summarised in Table 4.1.

It is interesting to observe that these median results for speed and comfort correlate well with the recorded entry and error rates. The two surface-aligned conditions (SUR2 and SUR10) received a median rating of 4 for speed and these were also the two fastest conditions in terms of entry rate. The 3 rating for condition MA2 and 2 rating for condition MA10 are also consistent with the entry rate trend observed in the quantitative data. Similarly, the accuracy rating of 4 for the two finger conditions (MA2 and SUR2) is consistent with the lower error rates observed in these conditions. The marginally higher error rates in condition MA10 compared with condition SUR10 is also consistent with the accuracy ratings of 2 and 3 respectively.

In terms of comfort, the on-table conditions were both perceived positively (median ratings of 4). By contrast the perception of comfort for two fingers in mid-air was neutral (3) and negative (2) for ten fingers in mid-air. This result is to be expected given the additional effort required in maintaining the hand cantilevered in space when typing in mid-air.

After completing both experimental sessions, participants were asked to select their preferred test condition. Note that participants were not made aware of their quantitative performance at any stage. The preference counts were 1, 14, 0 and 9 for each of MA2, SUR2, MA10 and SUR10 respectively. This result indicates a clear preference for the virtual keyboard plane being aligned with a physical surface. The subjective preference for the surface aligned conditions is also consistent with the quantitative entry rate results obtained. The most preferred condition overall was SUR2. This demonstrates good correlation between the subjective experience and quantitative performance given that SUR2 also yielded the highest mean entry rate and lowest mean error rate.

Table 4.1 Median response in post session survey on a Likert scale from 1-strongly disagree to 5-strongly agree. Questions asked participants to reflect on speed, accuracy and comfort in the typing condition.

Aspect	MA2	SUR2	MA10	SUR10
speed	3	4	2	4
accuracy	3.5	4	2	3
comfort	3	4	2	4

Table 4.2 Comparison of mean performance and behavioural measures for top and bottom performing users. Bold values indicate a significant difference based on an independent two-sample *t*-test at a 5% significance level.

Metric	MA2			SUR2			MA10			SUR10		
	Bot-6	Top-6	Diff.	Bot-6	Top-6	Diff.	Bot-6	Top-6	Diff.	Bot-6	Top-6	Diff.
wpm_{sim}	32.6	54.0	**65.8%**	41.2	82.0	**99.1%**	25.1	49.3	**96.8%**	35.0	69.5	**98.4%**
ER_{relax}	3.7	6.0	**62.9%**	4.0	4.4	10.7%	7.3	8.3	14.4%	7.8	6.7	-14.1%
IKI (ms)	387.8	232.0	**-40.2%**	314.7	154.9	**-50.8%**	495.7	270.6	**-45.4%**	369.1	188.5	**-48.9%**
Press Vel. (m/s)	0.49	0.62	26.5%	0.37	0.53	**42.6%**	0.30	0.48	**61.6%**	0.32	0.43	**32.3%**
Press Dur. (ms)	167.9	130.3	**-22.4%**	131.6	97.0	**-26.3%**	139.5	101.9	**-26.9%**	133.2	101.7	**-23.7%**
Press Depth (mm)	-14.0	-11.9	-14.8%	0.4	0.1	-71.5%	-2.8	-4.1	45.6%	1.7	1.6	-2.7%
Press Rev. (mm)	29.2	28.0	-4.1%	12.2	12.7	4.5%	23.1	25.2	9.2%	11.1	11.0	-1.0%

4.5.4 Indicators of High and Low Performance

In this section the key metrics that most clearly distinguish high performers from low performers in the typing task are examined. To this end, a comparison is made between the indicators of performance for the top six participants and the bottom six participants. Such an analysis can help highlight what constitutes 'ideal' typing behaviour.

The top and bottom six participants are each selected based on their mean entry rate across all conditions. Table 4.2 summarises the entry and error rates for these two groups in each condition and revisits several of the core micro metrics introduced and examined in Section 4.5.2.

As expected based on the group selection strategy, the top performers achieve significantly faster entry rates in all conditions. The top performers do, however, exhibit higher error rates in the MA2, SUR2 and MA10 conditions, with significantly higher error rates in the MA2 condition. This reflects a common performance trade-off of speed for accuracy.

The micro metrics presented in Table 4.2 highlight how the top group is generally faster in their movements. Table 4.2 suggests that the performance difference between the groups largely stems from shorter inter-key intervals and shorter press durations.

4.6 Implications for a Functional Keyboard

This section returns to the third objective of this study and the second part of *Research Question 1*: informing the design of a functional keyboard tailored for use in VR. Several examples of the way in which high-fidelity performance and behavioural data can inform the design of a fully functional keyboard system are reviewed.

First, as highlighted in Section 4.5.4, the behaviour of top performers informs tracking accuracy and 'touch' detection threshold target levels. For example, if the sub-group of top

performers is able to type in the vicinity of 80 to 100 wpm this implies a minimum detection threshold to make this feasible. For example, if an intersection based approach such as the one used in this study is used, the period between tracking position updates must be at least several times smaller than the typical press duration in order to avoid frequent failed detections.

Second, understanding the error distribution over the keyboard layout can inform likelihood estimates, that is, $P(key|touch_{x,y})$, in a probabilistic auto-correction strategy [192]. Similarly, understanding the specific performance of a given hand or finger can likewise inform the likelihood estimate, that is, $P(key|hand,touch_{x,y})$ or $P(key|finger,touch_{x,y})$. Such modulation of the confidence in particular touches might, for example, help to address the frequent insertion errors highlighted in Section 4.5.2 at the edges of the keyboard layout associated with the pinky fingers.

Third, understanding common errors can inform layout refinement and/or alternative 'touch' detection strategies. For example, frequent double-character omissions due to a failure to leave and re-enter the detection plane were observed. The analysis in Section 4.5.2 provides some preliminary insight into how such intent might be discriminated. For example, it may be feasible to detect the intent of a repeated character when a press reversal above a set threshold occurs while still inside the detection plane.

4.7 Discussion

This study highlights the complex nature of novel text entry system design. At the conception of this reported experiment, it was hypothesised that participant performance in mid-air with ten fingers would match, if not exceed, two finger performance. The results clearly indicate that the opposite is the case. The comparatively similar performance between the two surface-aligned conditions suggests that it may be hard for people to visually attend to more than two fingers on a virtual keyboard. There is precedence in this result with Kin et al. [80] determining that novice users employing two fingers (one per hand) can perform as well or better than when using ten fingers in a multi-target selection task.

The micro metrics introduced and examined in Section 4.5.2 form an attempt to shed light on the factors that dictate performance. Another perspective on this analysis comes from a brief consideration of the physiology of the hand and how this relates to typing. The physiology of the human hand means that movement of the middle, ring and pinky fingers can be difficult to decouple. The resistance provided in a physical keyboard is sufficient to prevent such coupled motion from generating insertions errors, however, there is clearly no resistance provided by a virtual keyboard in mid-air. Particularly problematic in ten-finger mid-air typing is the inability for users to decouple hand motion from fingertip motion. For example, a user may intend to

strike a key with their middle finger and move their wrist to do so but without a corresponding retraction of the ring and pinky finger this motion is likely to yield three distinct intersections with the detection plane. Intelligently addressing such errors represents a particularly difficult discrimination and inference challenge.

At this point it is also worth reflecting on the implications of the experiment protocol for the human performance envelopes that have been identified. Clearly a transcription typing task is very distinct from text composition. Vertanen and Kristensson [186] found that entry rates dropped by between approximately 15 and 35% depending on the nature of the composition task. When composing text other factors may dictate what features of a text entry system are preferable. Furthermore, the experiment exposes participants to extended periods of one-phrase-at-a-time text entry. This may not be representative of typical text entry use cases in VR. Despite best efforts to control for learning and exhaustion effects by fully balancing condition order, individual quantitative and qualitative results may inevitably be influenced by these factors.

Also of interest is a general sense of how the performance envelopes obtained compare with other studies conducted in this space. Walker et al. [189] and Grubert et al. [57] both examine the use of a physical keyboard with an HMD. Grubert et al. [57] found that entry rates on a physical keyboard were approximately 50% slower when wearing an HMD (with virtual representations of fingertips and keyboard) than when not wearing one. Participants in the study performed by Walker et al. [189] experienced only a marginal drop in performance when wearing an HMD but were supported by a probabilistic decoder.

4.7.1 Limitations and Future Work

There are several important limitations of this study and aspects of mixed reality typing requiring more detailed investigation. While the ultimate goal of this research is to develop highly efficient and easy to learn text entry methods for use with AR and VR HMDs, today's display and tracking technology necessitates an experiment conducted in VR. Current commercially available AR HMDs suffer from tracking, resolution and field-of-view limitations that were predicted to have a confounding effect on the investigation of raw performance potential. Nevertheless, many of the results obtained and behaviours observed are likely common to both deployments. Key differences, such as the effect of being able to see one's own physical hands as opposed to a virtual representation, require further investigation.

The use of the OptiTrack system meant that it was possible to ignore the limitations of conventional MR device-based tracking systems. As discussed in Section 3.4, this thesis is scoped to focus primarily on the boundary of the user interface exposed to typical application developers. It is important to highlight, however, that incorporating the uncertainty of a device-

based tracking system into the probabilistic reasoning around input for text entry would likely be fruitful. Research work is required to investigate the potential benefits of expanding the system boundary to incorporate such information.

It is also important to highlight that this study examines text entry without enforcing or requiring error correction. Clearly this fact means that the entry rate envelopes of human performance identified are optimistic. A fully functional text entry system for mixed reality must provide a means to perform error corrections and the best strategy for achieving this also requires future investigation. Also related to this point is the reduced keyboard layout used in this study. Again, a fully functional keyboard is likely to require the full complement of punctuation, numerals and support case modification.

For experimental purposes, participants in this study were confined to a seated posture. For any practical text entry system in mixed reality, however, a range of postures must ideally be supported. Posture may clearly have a strong influence on the performance and enjoyment of a given text entry strategy, and understanding this sensitivity remains as future work.

It is also important to highlight how certain design choices in the development of the test bed might necessitate caution in the generalisation of the results obtained. In the implementation examined, fingertips are represented as spheres instead of virtual hands. The effect of this choice has been examined by Grubert et al. [56] and Knierim et al. [83]. Grubert et al. [56] find that representing the fingertips alone can perform as well as live video of the user's hands with the added benefit of minimising keyboard occlusion. Note too that this presents a key distinction from potential performance in AR where users can see their own hands. Similarly, the choice of keyboard sizing and placement as well as placement of the input field may influence performance. For example, placement of the input field immediately above the keyboard potentially promotes focusing on keys rather than falling back on learned touch-typing skills. Likewise, some behaviours are potentially specific to keyboard layout, sizing and placement. The effects of these various design choices require future investigation.

A further avenue of future work is the expansion of the range of potential text entry strategies evaluated in the test bed. It is important to avoid design fixation and limiting investigation to those methods that are minor variations on conventional text entry strategies. For example, the virtualisation of the keyboard enables many novel input strategies such as split keyboards and/or keyboards that are fixed relative to certain joints. This does, however, expose a well-known factor in text entry research: the significant time investment required to learn a fundamentally new text entry strategy—and the fact that historically users have been unwilling to adopt a text entry method that demands upfront learning investment [88].

4.8 Conclusions

This chapter describes an empirical investigation of two fundamental design choices for text input in VR: the number of fingers engaged and whether the virtual keyboard is aligned with a physical surface or floating in mid-air. Aligning the keyboard with a physical surface is found to yield significantly higher entry rates, with greater user comfort. Contrary to expectations, the results suggest that users struggle to effectively leverage the availability of all ten fingers. In fact, when typing in mid-air, the availability of more fingers appears to be detrimental to performance. Nevertheless, the choice between surface or mid-air typing may be dictated by the user's circumstance and so it is useful to understand the anticipated envelopes of human performance of these different but complementary strategies.

In addition to identifying the envelopes of human performance for the four strategies investigated, this chapter also provides a detailed characterisation of the underlying features that distinguish each strategy in terms of its performance and behaviour. These insights in turn inform the design of a fully functional text entry system, including its tracking characteristics and statistical text decoder design. It is important to highlight that the conditions examined and distinctions made in their analysis are not a reflection of a desire to find a single 'best' input strategy for VR. Rather, in line with the overarching research methodology described in Chapter 3, it is hoped that a better understanding of the influence of various design decisions and underlying performance and behavioural indicators will ultimately yield a flexible and adaptable text entry system suitable for a variety of use-contexts.

4.9 Research Question 1 and the Design Process

This chapter investigated *Research Question 1: How can a designer obtain an understanding of the probabilistic characteristics of an interface; and, how can this understanding inform design in mixed reality?* The text entry use case examined serves as an illustrative example of how the probabilistic characteristics of an interface and its users can inform design. The use of a simulated decoder demonstrates how this understanding can be obtained with comparatively little developmental effort. Section 4.6 shows how the specific insights obtained through the characterisation can inform subsequent detailed design.

With reference to the emergent design process described in Section 2.3, this chapter serves as an example of Stages 1 and 2. The characterisation of the user and the system as part of Stage 1 was achieved by building a minimally featured test bed to simulate more advanced functionality to the user. Subsequent user testing as part of Stage 2 allowed the identification of the key determinants and feasible envelopes of performance.

The next chapter brings an alternative perspective to the characterisation problem by deriving design guidance through crowdsourcing. This is applied to the problem of adapting AR content to the user's context, with a specific focus on text presentation in AR. Later, Chapter 6 returns the focus to text entry and illustrates a more complete design process. It builds on this chapter by also demonstrating Stages 3 and 4 of the emergent design process: examination of sensitivity; and refinement and validation.

Chapter 5

Adaptation

A central design objective for next-generation mixed reality interfaces is the seamless melding of digital content into their deployment environments. Unlike most conventional HCI design problems, mixed reality is characterised by a complete lack of control over the physical context. The developer cannot reliably predict the range of physical environments in which their application will be deployed. This challenge is reflected by *Research Question 2* and the focus of this chapter: *How can a data-driven probabilistic preference model for the appearance of virtual content in mixed reality be efficiently obtained; and, how can this be leveraged to enable adaptation of mixed reality applications to uncertain deployment contexts?*

This chapter explores mixed reality interaction design conducted in the end-user's own context through crowdsourcing. The research question is investigated using a mobile web application that provides a guided MR experience while also facilitating the extraction of user context. This approach is applied to the challenge of dynamically adapting AR text content to the user's environment. Images of crowdworker contexts and subjective visual preferences given those contexts are captured to build a probabilistic preference model. Specifically, two key aspects of AR text content are examined: colouration and placement.

While secondary to the development of the model, this chapter also addresses the privacy concerns related to conducting in-context experiments that capture personal image data. Finally, Section 5.8 examines the opportunities, and improved external validity, afforded by large-scale deployment of web-based AR experiences in the development of emerging design guidance.

5.1 Introduction

The emergence of head-worn augmented reality represents an enormous opportunity for ubiquitous computing, potentially rivalling the dominance of the smartphone. With suitably compact and capable near-eye displays the smartphone screen becomes at best complementary and

at worst redundant. Despite the popularity of early examples of AR games and experiences, such as Pokémon GO [139], nascent AR designers lack the guidance, solution principles, and analytical approaches required to create aesthetic and seamless user experiences.

Compounding this lack of established principles for effective AR interface design is the fact that designing AR applications is considerably more difficult than designing smartphone applications. This additional complexity stems from three key factors: i) additional dimensionality; ii) unconstrained interaction spaces; and iii) unknowable deployment contexts. Research is required in all three of these areas. This chapter focuses on the third factor: unknowable deployment contexts, that is, the inability for the developer to foresee the environment in which their application will be deployed.

An example of the influence of context on AR interface design is the presentation of virtual text in the physical environment. Depending on the use case, the designer may wish this text to either subtly blend content with the physical environment or explicitly draw the attention of the user. Clearly an awareness of the user's physical context is necessary to deliver this behaviour.

A current obstacle to the derivation of generalisable and informative design principles for contextually adaptive AR is the limited availability of the head-mounted form-factor (for example, Microsoft HoloLens and Magic Leap). While researchers have demonstrated techniques with merit for commercial environments (e.g. [60, 37]), these demonstrations neither offer broader external validity nor do they illustrate practical techniques for developing more widely applicable design guidelines. Nevertheless, applications such as Pokémon GO have demonstrated that users can be engaged and immersed in AR experiences using simple smartphone-based low-fidelity approximations. Motivated by the increasing opportunities afforded by mobile crowdsourcing [74], this chapter hypothesises that crowdworkers might be similarly employed to gather information on AR deployment contexts and virtual-physical context dependence.

This chapter demonstrates how crowdsourcing can be leveraged to obtain a greater understanding of AR context dependence. A low-fidelity AR experience is deployed as a web application to prompt crowdworkers to capture images of their local environment while also obtaining feedback on the visual qualities of virtual elements overlaid on that context. The ubiquity of mobile devices and the increasing capabilities of web-based frameworks allow simple AR experiences to be quickly prototyped and rapidly deployed to a large number of users. This approach therefore allows large-scale testing and diverse dataset collection not afforded by lab-based studies.

The collection of image data from anonymous crowdworkers does expose potential privacy concerns. The method presented in this chapter accommodates these concerns by providing a user-driven obfuscation and acceptance protocol for sharing images. In demonstrating the

viability of this method, the investigation is thus conducted on two concurrent trajectories: 1) mediating the privacy concerns of crowdworkers; and 2) crowdsourcing AR deployment contexts and context-dependent data. In addition, the method is specifically employed in this chapter to an investigation of how virtual text content might be dynamically styled in AR given the physical setting. As a further illustration of the value of this approach, the understanding derived from this investigation is demonstrated in a high-fidelity head-mounted AR application. Therefore, the three key novel contributions of the chapter are:

1. A method for conducting AR experiments in the user's own context via crowdsourcing.
2. A protocol for mitigating the privacy concerns of crowdworkers as they share images of their local contexts.
3. A demonstration of the method in building a probabilistic preference model to enable dynamic adaptation of virtual content given background context in AR.

5.2 Related Work

The interplay between background context and virtual content presentation is a well-recognised challenge in AR [92]. From a technical rendering perspective, even ensuring virtual content colours appear as intended given the background is a non-trivial problem [106, 66]. More relevant to this work, however, is the challenge of selecting appropriate colouration of content *given* the background. A better understanding of this selection problem can enable dynamic adaptation of content appearance.

The legibility of text is particularly sensitive to appearance characteristics. The specific problem of ensuring text is legible in AR has been examined from various perspectives. A key step is understanding and predicting when text is not legible. Leykin and Tuceryan [99] trained a classifier aided by user evaluation to determine whether text overlaid on particular textures is readable. Similarly, Manghisi et al. [112] identified background texture qualities that determined whether text will be legible. Such knowledge is informative to interface designers but must be partnered with active strategies to change text appearance. As observed by Manghisi et al. [112], there are three distinct strategies for actively promoting text legibility in AR: i) adjust the text placement; ii) adjust the text appearance; and iii) place a panel behind the text. This first strategy of dynamic text placement has been widely explored in the literature [175, 176, 135, 136]. Rather than adapting the content, these implementations promote legibility by finding regions of dark, uniform texture on which to place text. However, many practical AR applications and interfaces are unlikely to support such free control over

the positioning of text. In many cases, it will still be necessary to fall back on the appearance adjustment and background panel strategies.

Gabbard et al. [48] examined three alternative schemes for actively modifying text colour in AR: complement, maximum HSV (hue, saturation, value) complement, and maximum brightness contrast. Their active schemes did not perform well, however, and a simple solution of blue text on an opaque white background panel (commonly referred to as a 'billboard') yielded the best performance. Gabbard et al. [47] subsequently find that maximising the luminance contrast ratio aids readability on billboards. Debernardis et al. [29] evaluated different presentation styles and billboard colours and suggest that white text on blue billboards yields good legibility. This result is reinforced by Kruijff et al. [91], who also examined preferences associated with the presentation of text labels in AR. Although evaluated on limited semi-static backgrounds, they find that blue background panels were overwhelmingly preferred. The findings of Kruijff et al. [91] correlated well with the maximum background colour brightness contrast. Interestingly, Albarelli et al. [3] performed a comparative evaluation between text with no background panel and text with a background panel. Based on their limited user study, they observe that no background panel was preferred or performed better in an assisted search task.

A common theme in the literature is the complexity of robustly accommodating diverse background textures and colours. Human perception capabilities and preferences are difficult to isolate in even the most tightly controlled psychological study. It is therefore unsurprising that small-sample HCI studies in this area uncover numerous perplexing results. While perhaps beneficial as preliminary guidance, the fact that a particular design, for example, white text on a blue billboard, has good general performance provides limited real insight to designers. It also ignores many implementation considerations, such as a desire to maintain a common aesthetic. Complicating this design problem further is the fact that, as observed by Ahn et al. [1], naïve approaches to dynamically updating visual properties of content can introduce flickering that is both distracting and annoying to users.

Crowdsourcing offers a potential strategy for greatly expanding the range of contexts in which perceptual issues are explored. Prior research has also previously demonstrated that crowdsourcing studies can replicate in-the-lab human perception studies [63]. There is also precedence [34] in applying crowdsourcing to facilitate interface feature design in mobile-based augmented reality.

5.3 Approach

This chapter demonstrates a crowdsourcing method for conducting AR experiments in the user's own context. Specifically, it explores how crowdworkers can be employed to distil design guidance for building contextually-adaptive text content in AR. The AR crowdsourcing method (described in detail in the following section) constructs a low-fidelity mobile-based AR experience that guides the user to collect spatially varied images of their environment. The user then adjusts the appearance of a text panel overlaid on the captured image. Two key sub-problems related to text panel presentation in AR are investigated in two separate experiments: Experiment 1 focuses on panel colouration; and Experiment 2 focuses on panel placement. The privacy concerns of crowdworkers related to sharing images of their private settings are also investigated concurrently as part of Experiment 1. Finally, the results from Experiments 1 and 2 are exploited to build a data-driven preference model for text panel design in AR. This model is demonstrated in a short validation study.

5.4 AR Crowdsourcing Method

The crowdsourcing method is based on a low-fidelity AR experience delivered by a mobile web application. The user's rear-facing camera stream is fed directly into the web page frame and virtual content is overlaid on this stream to deliver a through-the-screen AR experience. A web framework for building VR experiences[1] provides the functionality to ensure device movements produce corresponding changes in the virtual elements. Crowdworkers can then be instructed to perform specific activities or provide feedback on interface features in this setting. It is hypothesised that incorporating the crowdworker's own environment into the task promotes better engagement and more critical assessment of features.

There are two key advantages of this crowdsourcing approach over a conventional lab study. First, crowdsourcing facilitates the recruitment of a large and diverse participant group. This is important for the contextual adaptation use case as it helps provide good coverage over the range of background scenes. Second, such studies can be completed quickly and with relatively low cost compared with lab studies. This is particularly attractive when the goal of the research is exploratory. Qualifying these two advantages, however, is the lack of experimental control that can otherwise be exercised in lab studies. In general, there is always a balance to be struck between internal and external validity. Recognising these and other trade-offs, it is important to find ways to validate data obtained from crowdsourced user studies. By design, and as highlighted in the review of related literature, there exists

[1] A-Frame <https://aframe.io/>

Fig. 5.1 Distinct bird textures applied to the same bird model. From left to right: blackbird, blue jay, cardinal, canary, robin, wren.

a corresponding body of lab-based studies against which the crowdsourced results can be compared. The experiments conducted and their results are presented later in this chapter but first the established experimental framework is described.

5.4.1 Mobile AR Web Application

This investigation pursues a web-based architecture for two key reasons. First, online tasks are more readily integrated into existing crowdsourcing platforms. Second, a web-based implementation minimises the imposition on crowdworkers (that is, there is no requirement to install software) and removes friction in the steps between recruitment and completion.

To facilitate the capture of contextual information for the two experiments conducted, the application instructed participants to complete a series of simple target acquisition tasks. Participants located targets, styled as virtual birds (see Figure 5.1), that were presented at semi-random locations within their local environment. The rear-facing device camera stream provides the background of this virtual environment, producing a low-fidelity through-the-screen AR experience.

The virtual component of the AR scene was implemented using A-Frame. This framework helpfully manages the scene camera adjustment based on device orientation changes. It is important to note, however, that the framework does not currently support translation within the environment for mobile users: the position of the scene camera is permanently fixed.

The lack of registration between the physical and virtual scene also means that the AR experience is imperfect. Nevertheless, it remains sufficiently convincing for simple experimental and data collection tasks. The decision to frame the target acquisition task as an exercise in locating and photographing ‘birds’ mediates the disruptive effect associated with imprecision in the virtual-physical alignment. Participants may reasonably expect a bird to move around whereas this same behaviour may be more disruptive if the target is a fixed inanimate object.

The web application was deployed as a Human Intelligence Task (HIT) on the Amazon Mechanical Turk service. In order to commence the HIT, participants had to visit the listing using a mobile device. Upon accepting the HIT, participants reviewed a short description of the task and its purpose. This included the fact that images of their environment may be captured but would only be recorded after explicit approval from them. Participants were then required

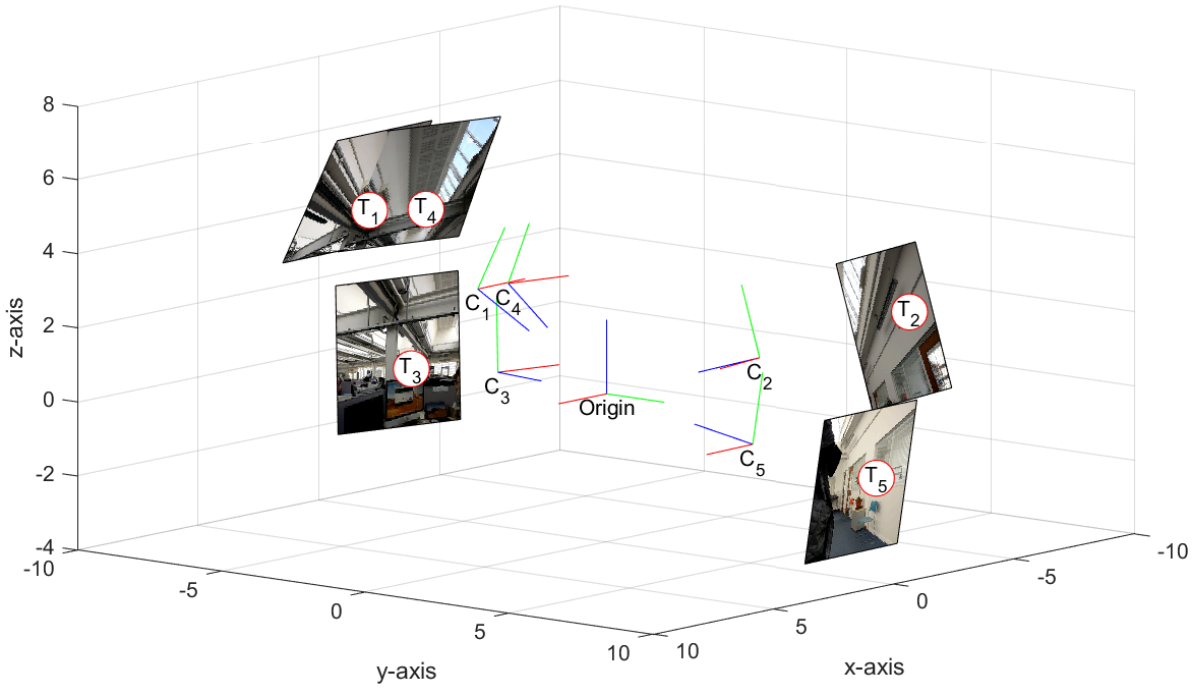


Fig. 5.2 Illustration of the spatial variation in the range of images captured around the participant's local environment. Targets are labelled T_n and camera orientations are labelled C_n where n is the sequence in which images were captured. Note that the camera frames, C_n, are shown with a z-offset from the origin to aid visualisation. This offset is not present in practice.

to explicitly express consent in order to complete the task. More detailed instructions were then provided on the role of the device camera and the image review process. The first bird capture activity was guided and then participants repeated the activity a further four times without explicit guidance.

The task phase of the AR web application involved three key stages: i) *image capture*; ii) *appearance refinement*; and iii) *image review*. The *capture* stage is described in the following section while the *review* stage is described later in the section addressing user privacy. The *appearance refinement* stage differs for the two experiments presented and so is described later in the context of the specific tasks performed.

5.4.2 Image Capture

The location of birds was quasi-randomised to promote spatial diversity in the context images captured from the participant's environment. For each instance of the target acquisition task (a participant performs 5 instances over the experiment), the new bird was located at between 60 and 100 degrees rotation from the current view azimuth. The sign of this offset was randomised.

The elevation of the bird was placed between -10 and 30 degrees elevation from the horizontal plane. An icon would appear at fixed time intervals in the centre of the participant's view to indicate where they must look to find the target. Figure 5.2 illustrates a single example of the spatial variation in captured images achieved through this dynamic target placement strategy. This figure shows how the placement of and guidance towards the target locations promotes the capture of environmental images from various perspectives.

Once the target is found, the participant must hold the reticle (mimicking the viewfinder of a camera) fixed on the bird. This serves two purposes: stabilising the virtual scene and ensuring captured images do not inherently suffer from motion blur. An animation of the reticle indicates when the bird is in focus. After the required focus period, the capture button is enabled. Participants then simply press the capture button and the background image from the rear-facing camera is temporarily recorded in memory on the client. It is at this point, with the image now recorded on the client side, that potential privacy concerns begin to emerge. These concerns and the mitigating solution applied are described in the following section.

5.5 Accommodating User Privacy

In developing the AR web application, it was hypothesised that privacy would be a key concern for crowdworkers. Although well placed to do so, there have been limited HCI efforts (e.g. [138]) to help users better understand and protect their online privacy. The challenge of supporting user privacy and accommodating concerns is exacerbated by the widely varying attitudes held by people about the disclosure of their personal details [75]. These concerns may additionally vary depending on context and the anticipated consumer, i.e. human versus autonomous agent. There is limited guidance available to designers as to the best strategy for maximising data capture while accommodating user specific concerns.

The privacy considerations in crowdworking have been examined from various perspectives. Daniel et al. [27] provide a survey of quality related issues in crowdsourcing and potential mitigation strategies. As an outcome of this survey, Daniel et al. define a quality model for crowdsourcing tasks which notably includes privacy as a potential factor influencing quality. Legion:AR [96] is a framework for augmenting activity recognition models by allowing crowdworkers to label uncertain cases while preserving privacy. The faces of people in the videos to be labelled by crowdworkers are obscured by auto-generated 'veils'. Beyond just individual privacy concerns, Lasecki et al. [96] suggest that reducing the resolution of video or image data is a reasonable strategy to avoid sharing sensitive information contained in the scene. The influence of blurring on the accuracy of crowdworkers performing behavioural coding of people in videos has also been investigated by Lasecki et al. [95]. These approaches

examine the preservation of privacy for people who appear in crowdsourced tasks but do not provide insight on how to manage the privacy of the crowdworkers themselves.

The objectives of McDuff et al. [116] and Tan et al. [174] are similar to this work in that they operate at the uncomfortable nexus of information capture and potential intrusions into privacy. McDuff et al. [116] solicited webcam footage of people watching commercials to generate a dataset of facial responses. Privacy was managed using an opt in approach. Participants entered the study through an advertised link and had to provide consent to enable access to the webcam. The consent rate for webcam access was 46.2% however this result is confounded by rejections stemming from incompatibility with the task, i.e. no webcam, or does not meet basic system requirements. Tan et al. [174] proposed a game suited to crowdsourcing for capturing user images in order to construct a diverse dataset of facial expressions. Its approach to dealing with privacy is to allow users to only send facial feature locations as opposed to raw images.

The literature suggests, therefore, two guiding principles of: i) limiting information capture to strictly what is necessary, and ii) giving users ultimate control over what is shared. In this chapter, these principles are applied in developing a privacy sensitive experimental protocol that has good generalisability beyond the specific investigation of context-dependence of textual content.

5.5.1 Image Review Protocol

To accommodate user privacy concerns, an architecture was chosen that ensured image data remained on the client side until it was approved. Only after approval would the image be sent to the server and saved in the database. Reflecting the hypothesis that workers would be generally unwilling to share personal image data, effort was taken to forestall the situation in which the majority of images were rejected. To this end, an obfuscation layer was included in the review protocol. Pixelation (also known as mosaicing) was the elected technique used for obfuscation. As part of the image review stage, the worker may increase or decrease the level of pixelation. To counter overuse of pixelation, the instruction given to users was to, “Please share as much image detail as you are willing.”

Pixelation was chosen for two key reasons: i) it is broadly familiar to a non-technical audience; and ii) it produces non-recoverable information loss. For completeness, it is important to note that pixelation is not necessarily a perfect de-identification method. If there is access to a database of images it may be possible to apply the same pixelation and match faces or objects [127]. In the absence of such a database, however, it is not possible for this kind of information to be recovered. Other advanced techniques (e.g. [156, 26]) could also be used to generate a depixelated image that is an estimated simulation of the original. While the raw detail cannot be recovered, it may be possible that the simulated image is sufficiently

Fig. 5.3 Illustration of the effect of increasing sub-block size, s, from left to right. No pixelation is $s = 1$ while $s = 20$ produces pixelation at 20×20 pixels.

realistic to raise the suspicions and concerns of a privacy conscious user. Addressing this type of concern among participants unfamiliar with the limitations of these techniques requires targeted investigation but is outside the scope of this current study.

In the *review* stage, the user may adjust the level of pixelation applied to the raw image by setting the sub-block size. Sub-blocks in the image are averaged and the average colour is used to replace all the pixels in the sub-block. Increasing the size of the sub-block removes more information from the image. This control was presented as a range slider with sub-block sizes: 1,2,4,6,8,10,12,16,20. Note that a sub-block size of 1 represents no pixelation. This range of values was chosen as they are factors of the default image resolution setting (480×480 pixels). Figure 5.3 illustrates the obfuscation achieved with a subset of the pixelation levels available.

After completing the *capture* and *appearance refinement* stages to the tasks, the participant was presented with the image *review* interface shown in Figure 5.4 (right). The default sub-block size upon presentation of the image review page was $s = 1$ (no pixelation). At this stage, the worker may choose to: a) approved the image; b) reject the image; or c) increase the level of pixelation and then approve the image. In practice, workers were far less concerned about sharing contextual images than hypothesised. Of the 2000 images captured by participants across both experiments, only 10 were rejected. Nevertheless, crowdworkers showed strong appreciation of the ability to reject and pixelate images with sensitive content. Detailed results are presented as part of Experiment 1.

5.6 Experiment 1: Panel Colouration

To make this investigation concrete, a text panel design use case was selected. Participants refine the appearance of a billboard-style virtual text panel appearing in the environment. When a bird is in focus (i.e. inside the view reticle), the bird name and a short description appear below on a coloured panel with 50% opacity (see Figure 5.4). Once the image is captured, participants are instructed to refine the appearance of the description panel.

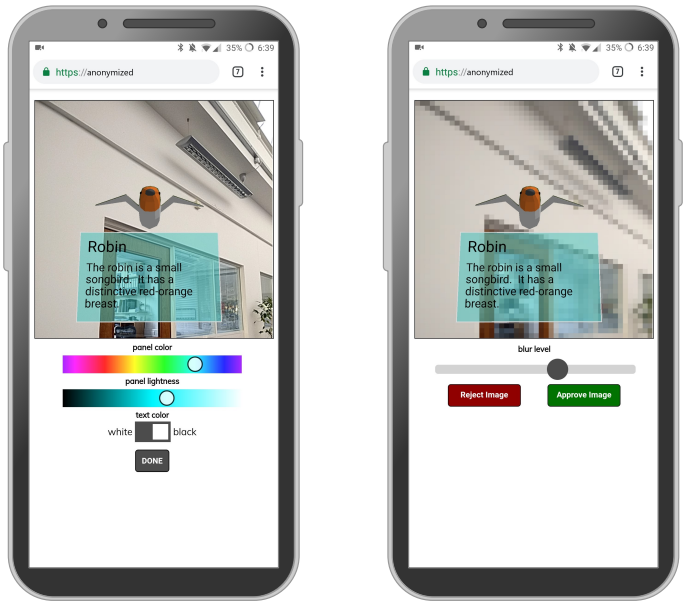


Fig. 5.4 The appearance refinement interface (left) allows the user to adjust the panel hue and lightness as well as choose between black or white text. The image review interface (right) allows the user to add pixelation to the image and accept or reject its transmission to the server.

To highlight the flexibility of this approach, participants were primed with the intentionally qualitative instruction, "Choose a colour that you think is best given the background. Please try to maximise visibility and readability of the text." This use case exposes an interesting and subtle interplay between subjective user impressions related to aesthetics and practical concerns relating to legibility. The interface for customising the description panel appearance is illustrated in Figure 5.4 (left). The top slider adjusts the hue while the bottom slider adjusts the lightness. The hue slider was initialised with a random rotation applied to the standard hue circle and the initial midpoint value was used as the initial panel colour. The lightness slider was always initialised to the midpoint value. A toggle is available to change the text colour between black and white. The toggle state was randomly initialised. The random initialisation of hue and text colour was done to prompt participants to make colour changes when required.

5.6.1 Results

A total of 200 participants (113 male, 84 female, 3 unspecified, 32.4 mean age) were recruited through Amazon Mechanical Turk for the study. The country-level location of participants is summarised in Table 5.1. Each received US$1 as compensation for their time. The mean completion time for the task was 8.5 minutes (including training and instructions).

Table 5.1 Participant locations in Experiments 1 and 2. For Experiment 1, 'Other' consists of one worker from each of Bahrain, Colombia, France, Mexico, Peru, Philippines, and Poland. For Experiment 2, 'Other' consists of one worker from each of Colombia, Dominican Republic, Ireland, Morocco, Pakistan, Peru, Poland and Romania.

Location	**Exp. 1**	**Exp. 2**
United States	159	127
India	14	19
Canada	7	7
Brazil	4	12
Indonesia	3	10
United Kingdom	2	2
Italy	2	2
Germany	1	5
Netherlands	1	2
South Africa	0	6
Other	7	8

Approval Rate and Participant Behaviour

With five task instances per participant and 200 participants there were a potential 1000 images to capture. The image approval rate was very high with only five images rejected in total by five different participants (an approval rate of 99.5%).

The degree of pixelation (sub-block size) was left unchanged in 54.5% of approved images. Recall that the sub-block range slider had a default initial value of $s = 1$ (no pixelation). In an additional 4% of images, participants raised the degree of pixelation before returning it to $s = 1$. The distribution of pixelation levels employed by participants is summarised in Table 5.2. Table 5.2 appears to show three distinct modes. The no pixelation default, $s = 1$, dominates (58.5%) but there is a second peak at $s = 6$ (6.8%) and a third peak at the other extreme, $s = 20$ (6.7%). This result suggests some stratification in the behaviour of participants. Additionally, a small number of the images appeared to be provided as extreme close-ups or with the camera lens covered, yet with no change to the pixelation level. This suggests that these users were further, conscientiously, trying to ensure their privacy. While this mechanism is entirely carried out on-device, these users may have been taking active steps to ensure images were not being surreptitiously captured without their consent.

The proportion of approved images in which the panel and text colour was altered provides a proxy measure for task engagement. The panel colour was adjusted in 75.7% of images and the text colour was adjusted in 47.6%. Note that a participant may not choose to change the panel colour if they consider its initial value to be appropriate given the background. Nevertheless, a

Table 5.2 Participant usage (%) of pixelation sub-block sizes, s, in Experiments 1 and 2. Note that no pixelation, $s = 1$, is the default and also the most frequently used setting. The usage results show three distinct modes at s =1, 6 and 20 across both experiments.

Exp.	**1**	**2**	**4**	**6**	**8**	**10**	**12**	**16**	**20**
1	58.5	3.1	4.2	6.8	6.5	5.6	5.1	3.4	6.7
2	54.0	3.7	5.6	6.6	6.1	3.7	4.8	3.4	12.0

very conservative estimate can be made that approximately three quarters of participants were actively engaged in the appearance refinement activity as instructed.

Billboard Colour Choice

This section describes the process of mining the collected context image and appearance refinement dataset for common patterns that inform the billboard colour selection problem. It is reasonable to anticipate that the dataset suffers from various noise factors, such as individual user preferences, user apathy[2], and interpretation differences. These types of factors appear in lab studies but their effect is more extreme in crowdsourcing due to the fact that only limited and unsupervised training can be provided in a HIT. Nevertheless, these effects can be combated by collecting large volumes of observations. To demonstrate the potential of a more complete dataset, this section shows that useful information, on par with similar lab studies, can still be extracted from the 200 participant dataset.

In addition to the noise factors described above, there are also aspects of the signal that frustrate simple analysis techniques. For an identical background context there are likely to be multiple billboard colour choices that yield similar legibility and aesthetics from the user's perspective. The concept of *hue templates*, for example, suggests that there are multiple alternative colour combining schemes that produce aesthetic results (see O'Donovan et al. [132] for an interesting discussion).

Accepting that the dataset likely contains both of these noise and signal effects, the analysis seeks to uncover any summative patterns reflected in the data. This analysis strategy involves; i) identifying informative groupings of similar background contexts; and ii) identifying common panel colours selected for these groupings. To do this, the sub-region or patch of the full image upon which the description panel was displayed is extracted. The dominant colour of this patch is then extracted by taking the mode of the hue histogram and the mean of the S and V values in HSV space. It is hypothesised that the patch hue is unlikely to influence panel colour selection

[2]A small minority of crowdworkers are known to race through tasks providing nonsensical data in order to minimise completion time [49].

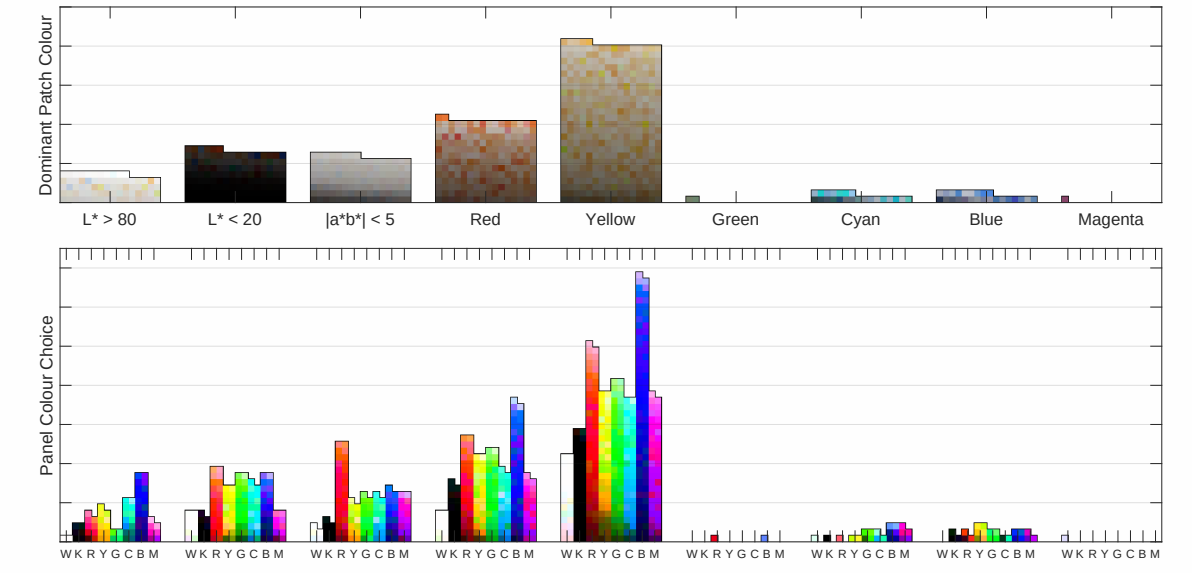


Fig. 5.5 Top plot shows a grouping of collected samples based on the dominant colour of the image patch with binning based on hue, lightness (L^*) and saturation (a^*b^*). Each pixel shows the dominant patch colour of an image sample. Notably, the full dataset contains comparatively few samples with a dominant patch colour of Green, Cyan, Blue or Magenta. Bottom plot shows the distribution of selected panel colours for each group, binned according to panel hue and lightness (W: $L < 0.1$, K: $L > 0.9$, R: red, Y: yellow, G: green, C: cyan, B: blue, M: magenta). A preference for blue and red panel colouration is observable, particularly in the Red and Yellow dominant patch colour groups.

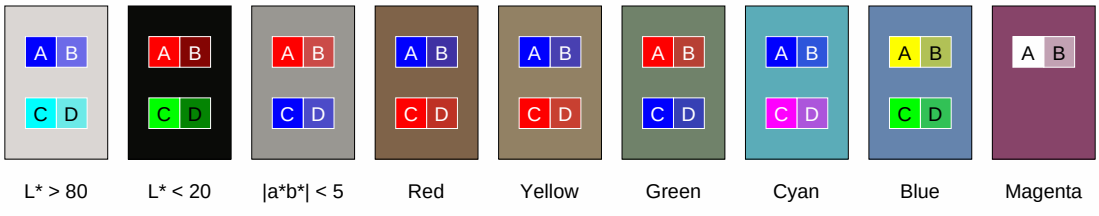


Fig. 5.6 Most (A) and second most (C) frequently selected billboard colour overlaid on the median colour of the corresponding groups in Figure 5.5. Billboards at 50% opacity shown by (B) and (D). Note that the Magenta group only includes a single sample and so no secondary billboard colour is shown.

at high (i.e. white) and low (i.e. black) lightness values and at low saturation (i.e. grey) values. To group on low saturation a threshold is placed on the vector a*b* (<5) of the dominant colour in CIE 1976 L*a*b* colour space. High and low lightness values are grouped by thresholds on L* (>80 and <20 respectively). The remaining ungrouped patches are then binned based on their hue value. Binning was performed according to standard 60° segments around the hue circle (with 'red' on the interval -30° to 30°). The resultant groups are illustrated in the top of Figure 5.5. Each patch is represented by a single pixel coloured based on the patch's dominant background colour.

The groupings show in Figure 5.5 are highly illustrative of typical background contexts to be encountered in AR. Review of the collected images indicates that the vast majority of images were captured indoors. Grey and white are common interior colourings. Similarly, the large groups for red and yellow correspond well with the large number of wood panelling and brick backgrounds captured. Far less common are background contexts with prominent green, cyan, blue and magenta colouring. The prevalence of black is largely due to images captured in low light.

The billboard colours chosen by participants corresponding to each of these background contexts were then grouped. To support summative review, the selected hue value was binned into its corresponding segment on the hue circle (again 60° segments, with 'red' on the interval -30° to 30°). Billboard colours with extreme lightness values (recall participants could modulate lightness) were separated into black (<0.1) and white (>0.9) bins. Figure 5.5 (bottom) shows the panel colour selection based on this binning. An interpretation, therefore, of Figure 5.5 is the distribution of billboard colour choice given the dominant background colour.

Figure 5.6 summarises the results of the background groupings by overlaying billboards of the most (A) and second most (C) frequent colour choices on the median colour of the clustered backgrounds. Also shown are the billboards at 50% opacity (B and D respectively). Clearly these groupings are sensitive to small datasets but the exploratory results are promising. Figure 5.6 shows a consistent preference for red and blue panels despite diverse background settings. Figure 5.7 shows a similar plot for grouping based solely on lightness. The corresponding most and second most frequent billboard colour choices are shown in Figure 5.8. These plot again highlight the general preference for blue panels, except when the background is very dark, in which case bright colours such as red and green are preferred. This result shows good alignment with the lab-based findings of Debernardis et al. [29] and Kruijff et al. [91] who found a distinct preference for blue panels. In contrast to these studies, however, a much better picture of the sensitivity of this choice is presented.

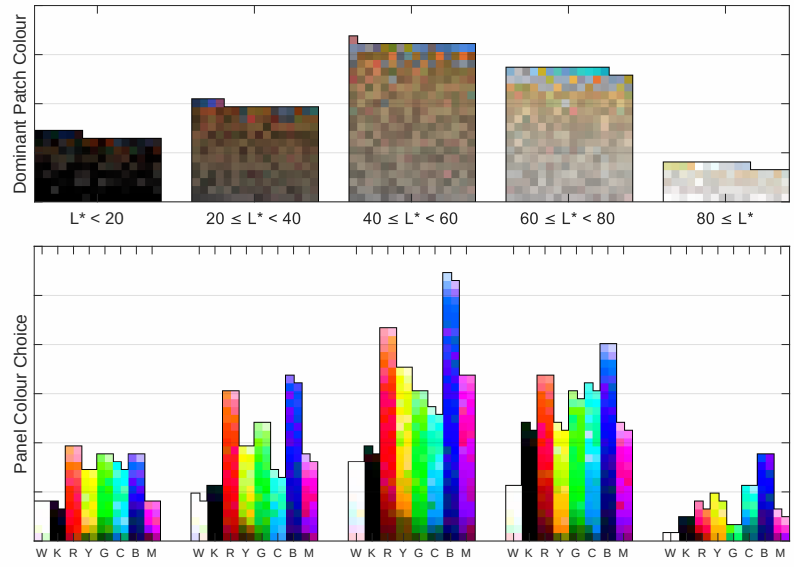


Fig. 5.7 Top plot shows a grouping of collected samples based on the dominant colour of the image patch with binning based on lightness (L^*) only. Each pixel shows the dominant patch colour of an image sample. Bottom plot shows the distribution of selected panel colours for each group, binned according to panel hue and lightness (W: $L < 0.1$, K: $L > 0.9$, R: red, Y: yellow, G: green, C: cyan, B: blue, M: magenta). Again, a preference for blue and red panel colouration is observable, particularly in the $20 \leq L^* < 40$ and $40 \leq L^* < 60$ dominant patch colour groups.

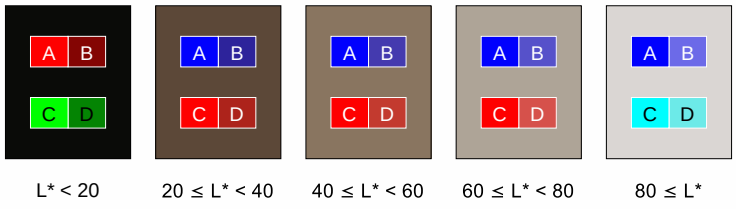


Fig. 5.8 Most (A) and second most (C) frequently selected billboard colour overlaid on the median colour of the corresponding groups in Figure 5.7. Billboards at 50% opacity shown by (B) and (D).

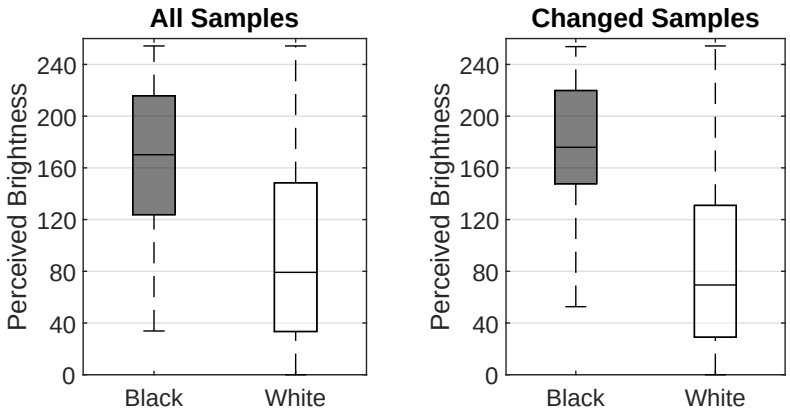


Fig. 5.9 Boxplots of the perceived brightness of the chosen billboard colour with grouping based on the user's selection of black or white text. The left plot contains all collected samples while the right plot contains only those samples where the text colour was changed by the user.

Text Colour Choice

The second critical aspect for text billboard design is the assignment of text colour. In the web application deployed, participants were allowed to toggle between black and white text. Therefore, the scope in this analysis is constrained to choosing between these two options.

The World Wide Web Consortium (W3C) provides a simple recommended formula for calculating the perceived brightness of a colour [150]. This yields a brightness value on the range 0 to 255. The W3C suggests a brightness difference of 125 promotes good visibility [150]. Intuitively, black text is more legible on bright backgrounds while white text is better on dark backgrounds. However, the web and its appearance characteristics are significantly different from those produced in AR so it is unclear how well this recommendation translates to environments with uncontrolled and potentially noisy backgrounds.

Figure 5.9 (left) shows the boxplots of perceived billboard colour brightness grouped according to the choice of black or white text in all samples. Figure 5.9 (right) shows the same boxplots but excluding samples in which the text colour was not changed. The median brightness for black text selection is significantly higher than that for white text selection as expected. The spread of each group does, however, highlight the fact that there is no clear threshold indicating the point at which one is clearly better than the other. Indeed, there is limited evidence-based guidance on an appropriate choice of this threshold. From Figure 5.9 (right) it can be observed, however, that the interquartile range of white text selection does not overlap with the interquartile range of black text. Therefore, the range between white text $q_3 = 131.0$ and black text $q_1 = 147.7$ may suggest a reasonable region of transition.

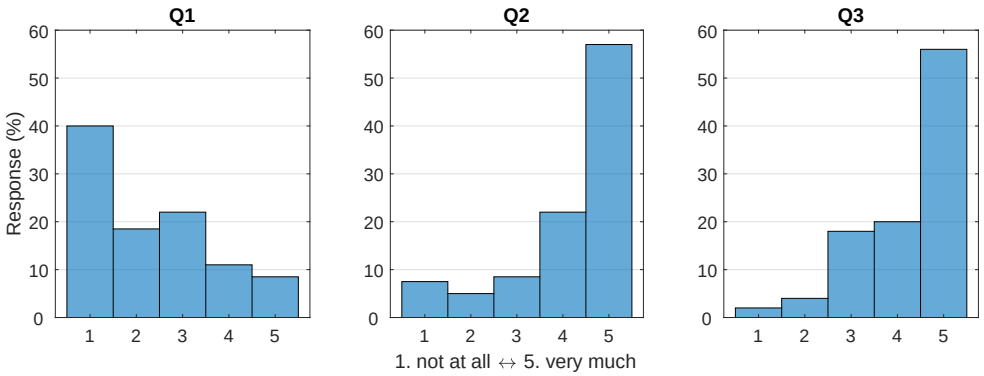


Fig. 5.10 Responses to survey questions from 1. not at all to 5. very much. Q1. Do you have any privacy concerns about sharing images of your workspace via a Mechanical Turk HIT? Q2. Do you think it is important to be presented with your images for review before sharing? Q3. Did you find the blurring capability useful for removing private detail from captured images?

Privacy Survey

After capturing the last image in Experiment 1, participants completed a short survey examining their privacy concerns. They were asked to respond to three questions on a five-point Likert scale. These questions and the allocation of responses to each are summarised in Figure 5.10. 58.5% of participants indicated that they were either not at all concerned or somewhat unconcerned about sharing images via a Mechanical Turk HIT from a privacy point of view. This result is remarkably consistent with the usage proportion of the default pixelation value. However, 79% of participants thought it was either somewhat or very important to be presented with these images for review. As a method for mediating privacy concerns it appears that the obfuscation capability was considered either very or somewhat useful by 76% of participants.

If a participant rejected an image they were asked to provide an explanation as to their main reason for rejection. For the five rejections among the 1000 total images in Experiment 1, the reasons offered were: sensitive information in the image (in two cases), family photos in the image, and lack of clarity (in two cases). It is interesting to note that the two images rejected due to the participants deeming them to be of poor quality reflects an eagerness to provide useful data rather than an express concern about privacy.

In summary, the findings related to privacy highlight that Mechanical Turk workers are generally willing to provide images of their local context. A majority are not overly concerned about privacy. The ability to obfuscate or reject sensitive images appears to successfully accommodate those with stronger reservations. As a strategy for maximising data acquisition while addressing participant concerns, the image review protocol presented appears to be very effective.

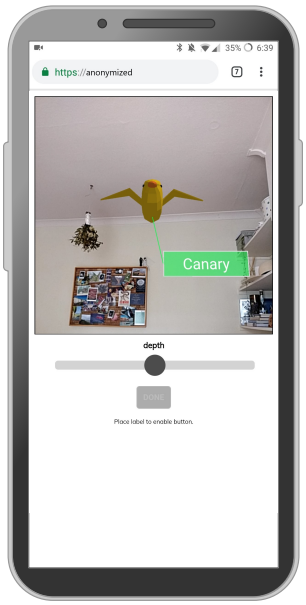


Fig. 5.11 The panel placement interface allows the user to adjust the label position by directly touching or dragging within the image region. The apparent depth of the label in the scene can also be adjusted using the slider.

5.7 Experiment 2: Panel Placement

Experiment 2 investigates the placement of text panels within the environment. Specifically, this experiment captured contextualised user feedback on the preferred placement of text panels, accounting for colouration, given the physical background. The process of capturing the initial image was identical to Experiment 1. Upon targeting the bird, however, rather than the full description panel shown in Experiment 1, only a label of the bird name was shown. This label was placed randomly around the bird but within the view frame and a leader line connected the bird model and the label. Once the image was captured, the participant was instructed to, "Place the label so as to maximise visibility and readability of the text." The label could be moved by simply touching on the screen within the image frame and/or by modifying the apparent depth of the label in the scene using a slider. The label placement interface is shown in Figure 5.11. Note that label colour was randomised and text colour was randomly assigned to be either black or white. Participants were still given the opportunity to review, reject or pixelate their images as required, however, the survey examining privacy concerns was removed.

5.7.1 Results

As with Experiment 1, 200 participants (125 male, 74 female, 1 unspecified, 31.2 mean age) were recruited through Amazon Mechanical Turk. Each received US$1 as compensation for

their time. Participants were only permitted to complete the task once (participants from Experiment 1 were not prevented from completing Experiment 2). The mean completion time for the task was 7.2 minutes (including training and instructions). The geographical location of participants in Experiment 2 is summarised in Table 5.1.

Approval Rate and Pixelation Behaviour

With each participant again capturing five images, there were a potential 1000 total images from 200 participants. Approval rate was again very high with only five images rejected in total by five different participants (an approval rate of 99.5%). The distribution over the usage of different pixelation levels is also roughly consistent with Experiment 1 (see Table 5.2). No pixelation, $s = 1$, again dominates (54.0%) but with secondary peaks at $s = 6$ (6.6%) and $s = 20$ (12.0%).

Label Placement Behaviour

Figure 5.12 illustrates the initial and final label placement centres. The target (virtual bird) is always centred in this window. Recall that the initial label location was randomised relative to the bird target. The left plot in Figure 5.12 clearly shows the random initialisation that places the label relative to the bird. The right plot in Figure 5.12 reflects the distribution of placement locations across all samples. Notable in this plot is the frequency of label placements above and below the bird model while also avoiding overlap with the model itself. This behaviour suggests a label placement preference that is, in part, independent of the background context.

It is interesting to test the hypothesis that a highly colourful background region will be avoided when placing the label. Hasler and Suesstrunk [62] introduce a simple *colourfulness* metric that can be computed based on an image's RGB colour space. Hasler and Suesstrunk [62] define the colourfulness metric, M, to provide correspondence with human judged attributes of an image ranging from *not colourful* to *extremely colourful*. Colourfulness, M, is computed using the formula:

$$rg = R - G, \tag{5.1}$$

$$yb = \tfrac{1}{2}(R + G) - B, \tag{5.2}$$

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2}, \tag{5.3}$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2}, \tag{5.4}$$

$$M = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb}. \tag{5.5}$$

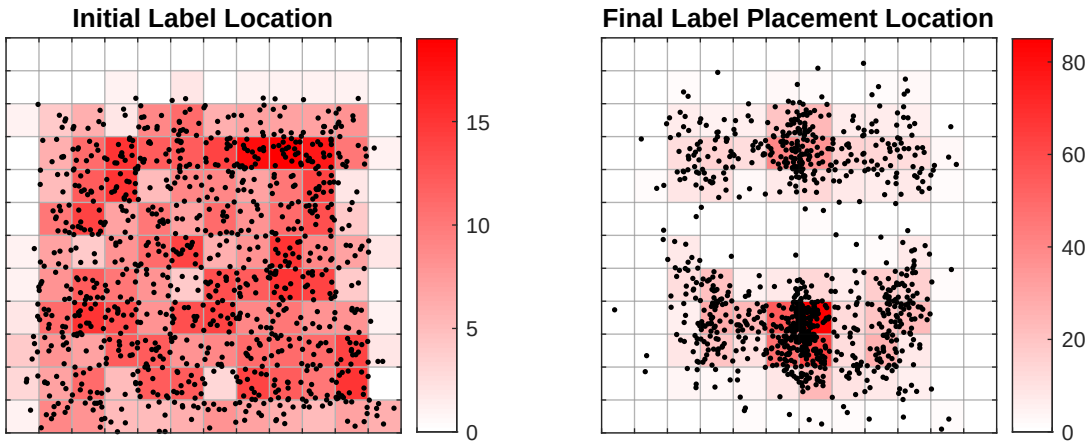


Fig. 5.12 Black dots denote the initial (randomised) label placement location (left) and final label placement location (right) within the captured image window. Frequency of placement within image sub-regions (regular 40×40 pixel blocks) is represented by the colouration.

Figure 5.13 shows boxplots of the change in colourfulness, ΔM, between the initial label placement region and the final label placement region for three groups of initial region colourfulness. The change in colourfulness, ΔM, will be negative when the label is moved from a colourful region to a less colourful region. Figure 5.13 suggests that when the initial region is *not colourful* ($M < 15$), users typically find a region that is similarly flat in colour. When the initial region is *slightly colourful* ($15 \geq M < 33$), there is some sign of a general preference for placement in regions yielding a negative ΔM. When the initial region is *moderately colourful* or more ($M \geq 33$), there is a definite bias towards a negative ΔM. This suggests that less colourful regions are preferred for label placement.

Another informative point of analysis is the influence of background clutter on label placement. *Edgeness per unit area*, F, is a simple metric for quantifying the degree of texturing or 'busyness' of an image [172]. F is computed for a region of N pixels by counting the number of pixels, p, for which the gradient magnitude, $\mathrm{Mag}(p)$, exceeds threshold, T. More concisely:

$$F = \frac{|\{p|\mathrm{Mag}(p) \geq T\}|}{N}. \tag{5.6}$$

The change in edgeness between the initial label patch and the final label patch, ΔF, provides an indication of the effect of background 'busyness'. Figure 5.14 shows boxplots of ΔF over three groupings of initial patch edgeness ($T = 100$). Moving from a patch with high edgeness to a patch with less texture will yield a negative ΔF. Figure 5.14 suggests that when the initial patch has low edgeness (<5%) the ΔF is likely to be close to zero. As the edgeness of

the initial patch increases, however, participants increasingly relocate the label to less textured regions (yielding a negative ΔF).

In summary, Experiment 2 highlights several key determinants of label placement preference: offset, colourfulness and edgeness. Recall that users were unable to set the panel colour in Experiment 2 and so placement related concerns are expected to dominate. Clearly, the preferred placement location may be influenced by the panel colour and this interaction requires investigation. Nevertheless, in instances where a billboard can be dynamically placed, it is advantageous to select a preferred region before adjusting the colouration.

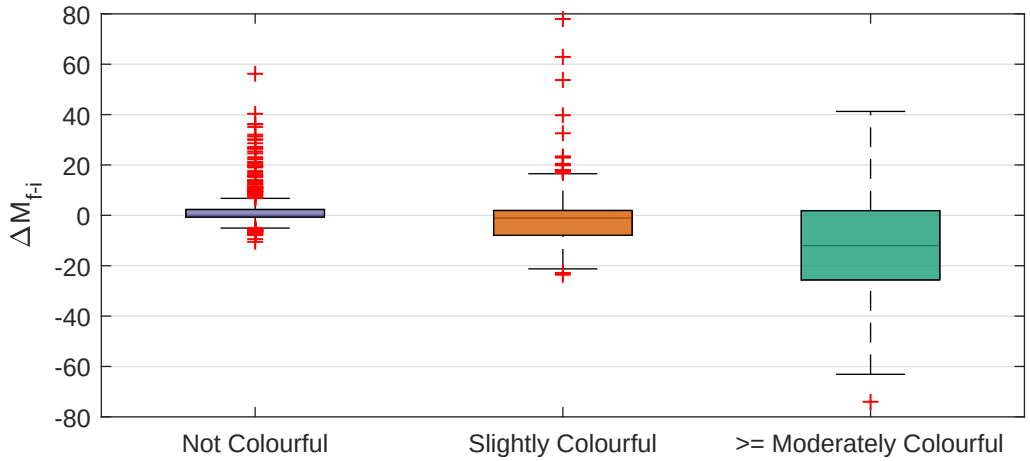


Fig. 5.13 Boxplots of change in *colourfulness*, ΔM, between initial and final label placement region. Red crosses indicate outliers based on $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$.

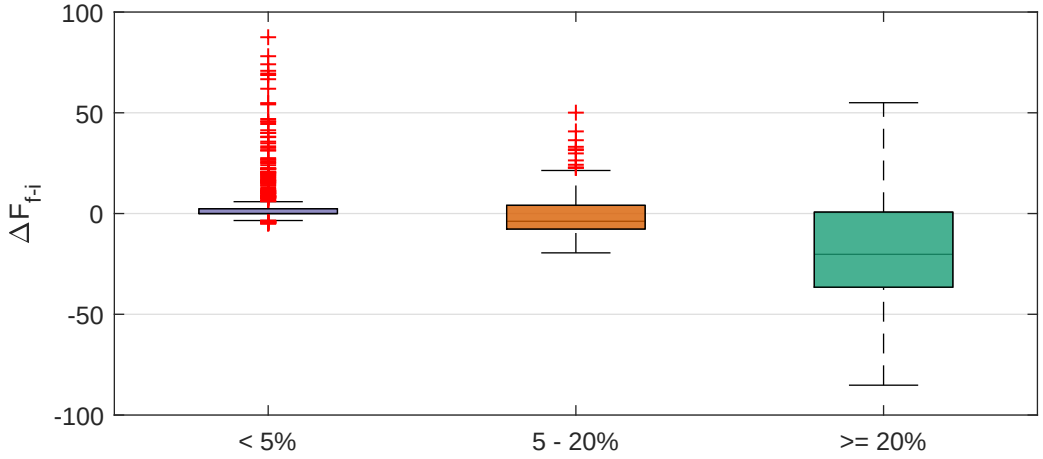


Fig. 5.14 Boxplots of change in *edgeness per unit area*, ΔF, between initial and final label placement region. Red crosses indicate outliers based on $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$.

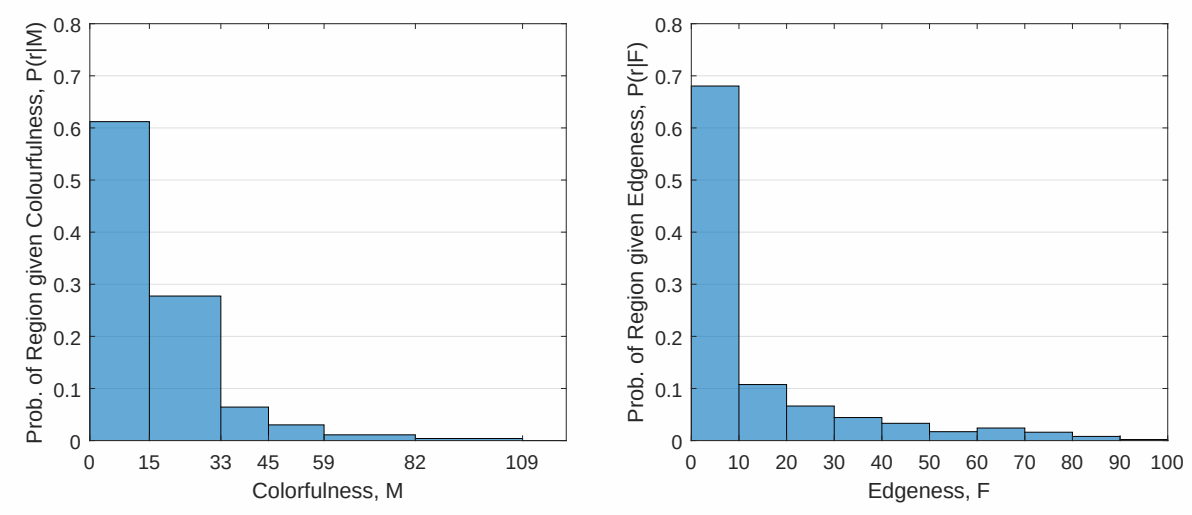


Fig. 5.15 Estimated probability distributions of the likelihood of sub-region selection given colourfulness, M, (left), and edgeness, F (right). Note that the binning of colourfulness, M, is based on the groupings defined by Hasler and Suesstrunk [62].

5.8 Validation Study: Dynamic Text Panels

The purpose of the Validation Study is to highlight the viability of the described AR crowdsourcing experimental method. To confirm the design guidance obtained is useful and implementable, a solution for contextually adaptive text panels is demonstrated in a high-fidelity AR application. This application, designed for use with the Microsoft HoloLens, provides dynamic placement and colouration of billboard style *tooltips*. The performance of this dynamic tooltip functionality is compared with a non-dynamic baseline in a short user study. The design of the dynamic text panel procedure derived from the collected data is now presented.

5.8.1 Design

Formalising the colour selection and placement problem for text panels in AR necessitates the consideration of three sub-problems: i) billboard colour choice; ii) text colour choice; and iii) billboard placement. A simple strategy for dynamic text appearance adaptation can be derived from the collected data using a compounding probabilistic approach. The approach converts the frequency responses observed for colour choice and placement (in terms of offset, colouration and edgeness) into probabilities. It then combines them to yield a mixture distribution estimating the preferred placement sub-region, r, in an image and the preferred colour, c, given that region. The estimated distributions for placement region given colourfulness and edgeness are presented in Figure 5.15. The estimated distribution for placement offset (normalised based

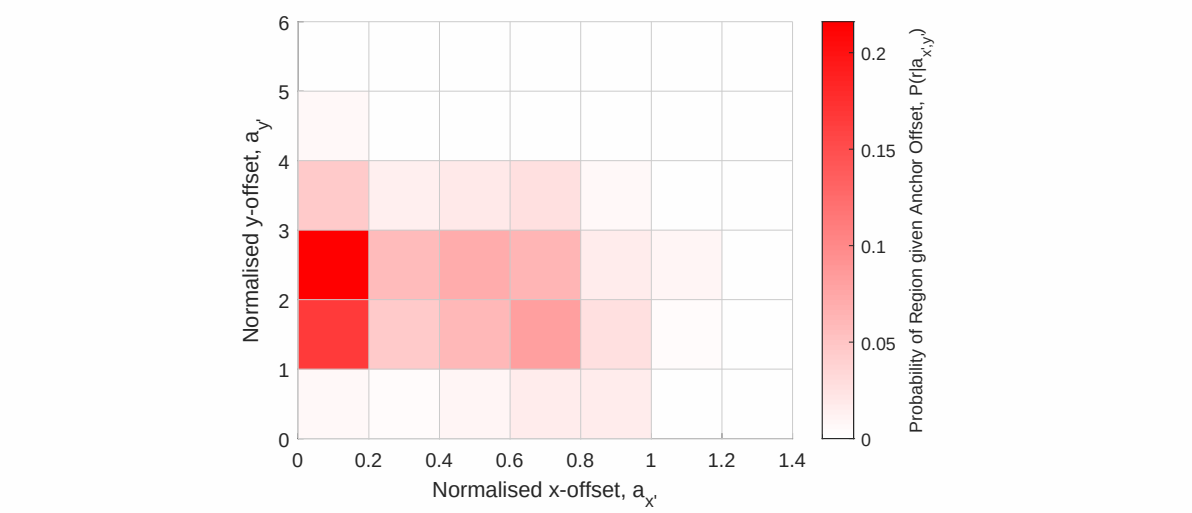


Fig. 5.16 Estimated probability distribution of the likelihood of sub-region given normalised x, y offset. Note that the normalised offset is taken to be symmetric about the x and y axes.

on the billboard width for x and height for y in image coordinates) is presented in Figure 5.16. This procedure requires transforming the tooltip anchor location (i.e. the point to which the tooltip is to refer) into the image coordinate system and the selected tooltip location in image coordinates back into the world frame.

Accepting too that designers typically wish to provide an interface with a consistent colour palette, a final uniform distribution is applied over a set of predetermined colours. This distribution serves to bias the colour selection towards selecting only from within the palette, but informed by the preference model. The steps involved in delivering this dynamic tooltip functionality are summarised in Algorithm 1.

The estimated likelihood of selecting region, r, given edgeness (i.e. line 5) for an example tooltip target location in this study is illustrated in Figure 5.18. The combined mixture distribution for this same target location (i.e. summing log probabilities at line 7) is illustrated in Figure 5.19. The resulting tooltip placement and colouration for this target location is then illustrated in Figure 5.21.

A three colour palette (see Figure 5.17) was selected for the study using an online colour scheme generator (using the triad scheme at paletton.com). This scheme was chosen to reflect

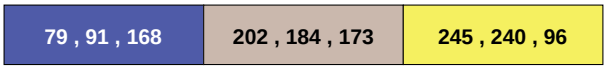


Fig. 5.17 Selected colour palette for constraining the dynamic tooltip colour assignment in accordance with an established aesthetic.

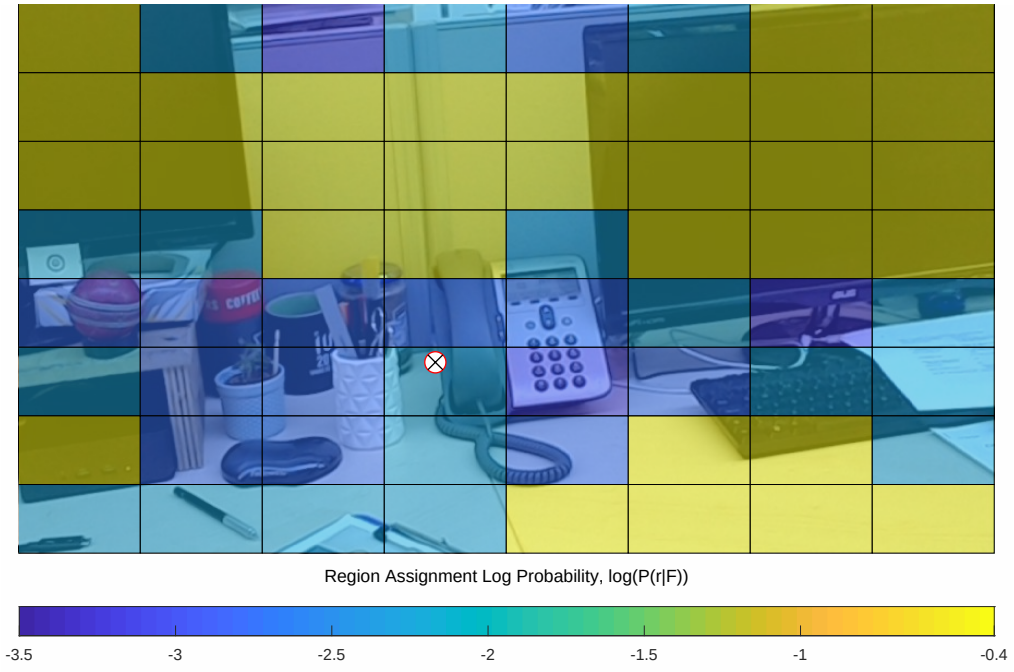


Fig. 5.18 Tooltip placement probabilities given edgeness of sub-regions in the background image. The tooltip anchor centre is indicated by the red circle.

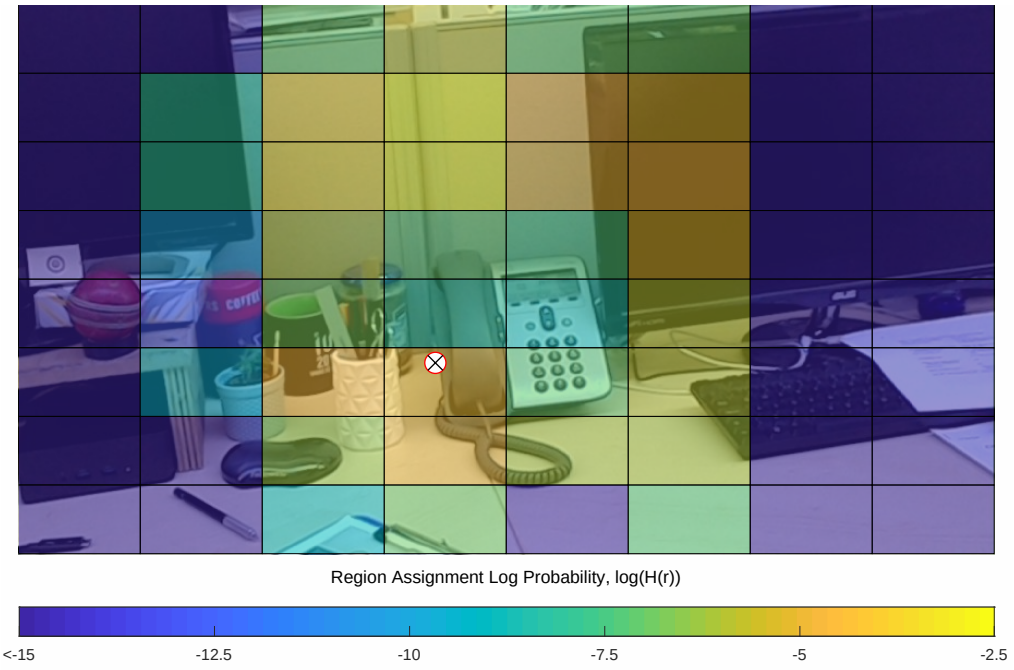


Fig. 5.19 Resultant tooltip placement mixture probabilities of sub-regions in the background image. The tooltip anchor centre is indicated by the red circle.

Algorithm 1: Contextually Adaptive Tooltips

Function `AdaptTooltip(`I, a`)`
  Input : Background image, I, and tooltip anchor location, $a_{x,y,z}$
  Output : Tooltip position, $t_{x,y,z}$, billboard colour, c_b, and text colour, c_t
  Transform anchor position, $a_{x,y,z}$, into its equivalent position in image coordinates, $a_{x',y'}$
  foreach *Sub-region, r, of the background image, I* **do**
    Lookup $P(r|M)$, i.e. probability of selecting r given colourfulness, M
    Lookup $P(r|F)$, i.e. probability of selecting r given edgeness, F
    Lookup $P(r|a)$, i.e. probability of selecting r given offset from anchor position, $a_{x',y'}$
    Combine $P(r|M)$, $P(r|F)$ and $P(r|a)$ to yield mixture distribution, $H(r)$
  end
  Choose sub-region, r_{max}, corresponding to the maximum of the mixture distribution, $H(r)$
  Extract dominant patch colour, c_p, and patch lightness, l_p, from image region r_{max}
  foreach *Billboard colour group, g, in billboard colour groupings* **do**
    Lookup $P(g|c_p)$, i.e. probability of selecting g given patch colour, c_p
    Lookup $P(g|l_p)$, i.e. probability of selecting g given of patch lightness, l_p
    Lookup $P(g|palette)$, i.e. probability of selecting g given defined colour palette
    Combine $P(g|c_p)$, $P(g|c_p)$ and $P(g|palette)$ to yield mixture distribution, $G(g)$
  end
  Choose group, g_{max}, corresponding to the maximum of the mixture distribution, $G(g)$
  Choose billboard colour, c_b, corresponding to g_{max} in palette
  Choose text colour, c_t, based on threshold of the perceived brightness of colour c_b
  Transform image coordinates of the centre of r_{max} to equivalent world position, $t_{x,y,z}$
end

the general preference for red and blue observed in Experiment 1. A pale red was selected to reflect the preferred use of red on dark. Yellow completes the colour scheme (note this was the only other colour achieving the most preferred rank in Experiment 1, excluding white on magenta for which there was only one sample).

5.8.2 User Study

The dynamic tooltip functionality described was evaluated in a short user study with 8 participants (1 female, 7 male). Participants were instructed to locate physical objects in the environment, to which virtual tooltips are attached. For experimental control purposes, participants were instructed to remain seated while completing the task. Once a tooltip location is found, the participant must focus on the tooltip anchor for 5 s before it is displayed. This delay serves two purposes: i) it steadies the participant's head and primes them for the subsequent text scanning task; and ii) it provides an opportunity to perform the dynamic text panel assignment.

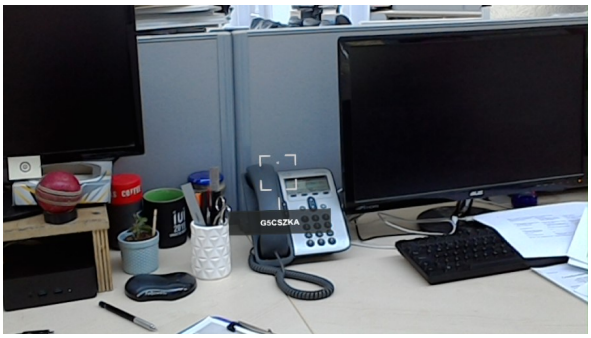


Fig. 5.20 Tooltip placement and styling in the BASELINE condition.

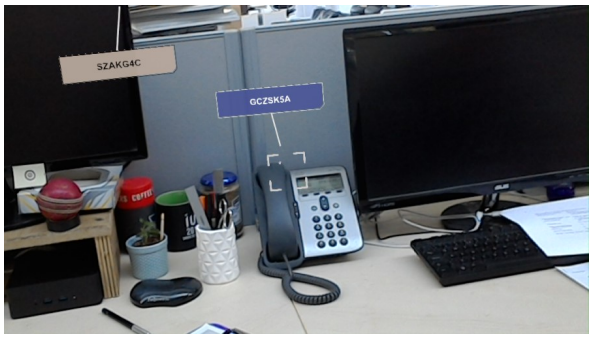


Fig. 5.21 Tooltip placement and styling in the DYNAMIC condition.

When the tooltip is displayed, the participant scans the text to locate a single digit among a collection of pseudo random characters. The user enters this digit on the keyboard and the elapsed time since tooltip display is recorded. This task is derived from that proposed by Gabbard [46]. Participants perform the task in two conditions, BASELINE and DYNAMIC, with eight tooltips per condition (order of conditions was balanced over participants).

The BASELINE condition places panels randomly directly above, below, left or right of the anchor location. Colouration is allocated randomly (though ensuring 4 of each colour) from two standard materials used widely in HoloLens example applications (MRTK): dark grey and blue, both with white text. The DYNAMIC condition produced tooltips using the procedure described. For convenience of development, the assignment procedure ran remotely on a server with the image sent over a dedicated wireless network and the corresponding tooltip location and appearance details returned to the HoloLens. Examples of tooltips produced by each condition for the same participant at the same target location are shown in Figures 5.20 and 5.21.

5.8.3 Results

The quantitative results presented in this section should be interpreted with a degree of caution given the limited number of participants involved. A comparison between the mean participant reaction time results in each condition is informative as to the viability of the procedure presented. Figure 5.22 presents boxplots of the mean reaction time for each participant in each condition. The median reaction times of the participant group were 1673 ms and 1297 ms for BASELINE and DYNAMIC respectively. This suggests a marginal reduction in reaction time but more importantly, that the DYNAMIC procedure delivers behaviour at least on par with the BASELINE approach. Figures 5.20 and 5.21 also provide a useful illustration of how this

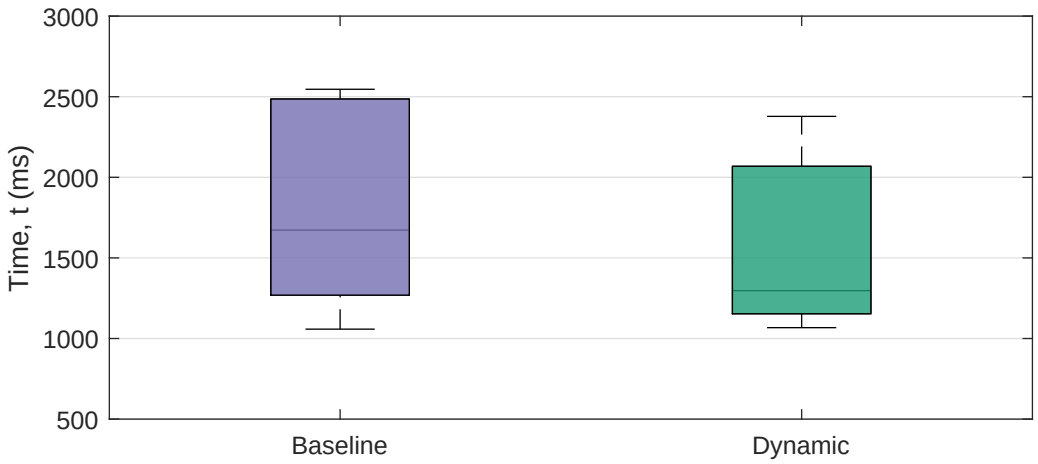


Fig. 5.22 Boxplots of mean participant reaction time. The median reaction times of the participant group were 1673 ms and 1297 ms for BASELINE and DYNAMIC respectively.

functionality successfully selects a tooltip location and appearance that, at least for this single case, is preferable to the naïve approach.

In summary, these results suggest that there may be value in the dynamic tooltip procedure described. A more complete experiment is required to definitively test this hypothesis. The aim in this section, however, is to validate the AR crowdsourcing method and illustrate how collected data can be operationalised. Although provisional, these findings do satisfy that aim.

5.9 Discussion

This chapter serves as a vehicle for demonstrating the value of crowdsourced AR evaluation datasets. In this investigation, context and physical-virtual dependence information was captured across heterogeneous settings. This information was readily operationalised to build a prototype application delivering contextually-adaptive text content on an early commercial AR HMD. Several limitations of the investigation and promising avenues of future research are discussed below.

5.9.1 Limitations

A limitation of this work is the confined set of design controls given to users for changing the description panel appearance. Only billboard hue, luminance and text colour could be varied in Experiment 1, while only panel placement could be modified in Experiment 2. This constrained interaction space was designed so as not to overwhelm users and to avoid excessive ‘twiddling’

behaviour. However, future work will be necessary to examine how additional design control can be provided to users without these disruptive effects.

The pass-through AR approach demonstrated, while readily available using mobile devices, may not be suited to all types of augmented reality experiences. For example, mobile AR does not face many of the additive colour effects introduced by optical see-through AR. However, as demonstrated, other aspects of a particular design problem, such as colour blending and contrast, could be rapidly prototyped. Some specific use cases may also not be as suitable to this type of low-fidelity AR implementation. One such example might be non-intrusive persistent notifications, where a fixed user-relative position (rather than world-fixed position) may be desired.

The perception of colour is sensitive to a great number of factors. In this investigation, no calibration or specific display settings on the device were enforced. This means that colour rendering differences between devices may introduce noise into the user feedback. The experimental choice was made to not control for this factor since this is more realistic of an actual user's experience with a simple application: it is useful to have a model for dynamic adaptation that works for most users in most cases. Nevertheless, there is an opportunity for a strict investigation of how device variation might influence design choices and what experimental controls can be applied to address this factor.

Another important point of discussion related to privacy is the influence of the nature of the task, and the affiliation of the task requester. The image capture aspect of the task was intentionally embedded within the bird finding activity. The plausible reason for capturing context images was designed to positively bias participants towards the task, but was not specifically investigated as a factor. Related to this is the task introduction, which made explicit mention of the university affiliation. As highlighted in one of the participant comments, tasks requested by researchers (affiliated with a well-known institution) may again positively influence participants. Such aspects of participant behaviour are difficult to quantify without running studies that are intentionally misleading (e.g. [73]).

5.9.2 Future Research Opportunities

There is active work in streamlining mobile AR frameworks for use in the web. The WebXR Device API[3] is a working draft for supporting VR and AR on the web. This standard outlines support for 6 degrees of freedom (DOF) pose tracking with mobile devices. This presents a significant opportunity for enhancing the fidelity of the mobile AR experience presented to crowdworkers. With 6DOF tracking, the experiments described in this chapter could examine

[3]https://www.w3.org/TR/webxr/

a wider range of additional factors influencing label placement specifically and AR content presentation more generally.

While the example experiments were limited to static image capture, the approach, where sufficient bandwidth is available, could be extended to real-time video capture and processing. This would enable the investigation of temporal and spatial coherence of virtual content.

Lastly, these preliminary results motivate the collection of a more extensive dataset as future work. However, to make such datasets available to the HCI and AR community it will be necessary to deal with still more complex privacy concerns, especially around image release. In the images collected as part of this study there were a total of 10 images that captured people who, based on the orientation of the image and scene, were not the main participant. Furthermore, there were a number of images containing documents or photos as well as outdoor locations that might enable identification of individuals. There is careful research work required to ensure streamlined methods of data capture, such as the methodology proposed, do not inadvertently leak identifying information to public datasets. Furthermore, as highlighted in Section 5.5.1 there are ethical issues to be addressed around making users fully aware of the effectiveness and limitations of various image de-identification techniques.

5.10 Conclusions

This study demonstrates that crowdsourcing context information for adaptive AR is not only feasible but also efficient. For only US$440 (including commission), two context-dependent AR user studies were conducted with 400 users spanning 22 countries to assemble a dataset of almost 2000 images and user-defined billboard preference profiles. Crowdworkers were willing to engage with a low-fidelity AR experience and to share images of their local environment. Overall, this chapter highlights new avenues for investigating and evaluating contextually-informed AR applications using crowdsourcing. Considering the inherent complexity in AR user interface design, crowdsourcing is a promising complementary method to assist evolving new data-driven designs which are difficult to achieve using traditional lab studies. The potential improvements in external validity offered by the AR crowdsourcing method for obtaining emerging design guidance is a crucial contribution at this early stage of AR development.

5.11 Research Question 2 and the Design Process

The focus of this chapter is the investigation of *Research Question 2: How can a data-driven probabilistic preference model for the appearance of virtual content in mixed reality be ef-*

ficiently obtained; and, how can this be leveraged to enable adaptation of mixed reality applications to uncertain deployment contexts? This chapter demonstrates an effective strategy for building a probabilistic preference model through crowdsourcing. This strategy is shown to be both efficient and effective. The second part of the question is examined in Section 5.8. The ability of the derived model to adapt content to the deployment context is demonstrated in an illustrative use case.

With regard to the emergent design process described in Section 2.3, the approach described in this chapter is a further example of efficient characterisation of the user and the system (Stage 1). Crowdsourcing delivers this outcome with good efficiency and at relatively low cost. The data in turn enables the implicit encoding of the determinants of performance (Stage 2) into the probabilistic preference model. At this point, the model is already capable of delivering useful application behaviours but further refinement and validation could be performed in line with the subsequent Stages 3 and 4.

Chapter 6

Inference

This chapter explores *Research Question 3: How can probabilistic inference be exploited to accommodate high levels of input noise in mixed reality applications to deliver more efficient interactions?* Continuing the theme from Chapter 4, this case study also investigates text entry but with a particular focus on augmented reality. The specific focus on augmented reality is interesting in this context given the high levels of input noise typical of currently available AR head-mounted displays.

In examining the research question, this chapter presents the VISAR keyboard: a text entry system tailored to AR head-mounted displays supporting error-tolerant input via a virtualised input surface. Users select keys on the virtual keyboard by imitating the process of single-hand typing on a physical touchscreen display. The system uses a statistical decoder to infer users' intended text and to provide error-tolerant predictions. There is also a high-precision fall-back mechanism to support users in indicating which keys should be unmodified by the auto-correction process. A unique advantage of leveraging the well-established touch input paradigm is that the system enables text entry with minimal visual clutter on the see-through display, thus preserving the user's field-of-view.

This chapter describes the process of iterative refinement and evaluation of the system, with the final iteration of the VISAR keyboard supporting a mean entry rate of 17.75 wpm with a mean character error rate less than 1%. This performance represents a 19.6% improvement relative to the state-of-the-art baseline investigated: a gaze-then-gesture text entry technique derived from the system keyboard on the Microsoft HoloLens. Finally, Section 6.10 validates that the system is effective in supporting text entry in a fully mobile usage scenario likely to be encountered in industrial applications of AR HMDs.

6.1 Introduction

Recent progress in head-mounted displays (HMDs) for augmented reality (AR), such as the Microsoft HoloLens, demonstrates the commercial potential of AR to support new forms of interaction and work in a range of industries including construction, education and health. Text entry is an integral activity in such AR environments, allowing users to, for example, send short messages, annotate physical objects and digital content, compose documents or fill out forms. The placement of the user within a virtually augmented environment introduces new and exciting opportunities for the interface designer. The design space is considerably broadened by the additional dimensionality available and new forms of interaction are made possible.

New challenges also emerge in providing effective text entry for AR HMDs. First, currently available devices are typically limited in terms of their display region size. Compounding the limited size is the fact that the display region is located in the centre of the user's field-of-view. Delivering a text entry method that preserves field-of-view while supporting effective input presents a unique design challenge. Second, delivering immersive and fully mobile AR applications in which the user can freely explore and interact with both the physical and virtual environment suggests avoiding input devices that encumber the user. Avoiding encumbering the user while maintaining freedom of mobility means that external (off-body) sensing to support text entry is also not practical. Third, a key goal of AR applications in general should be to minimise or eliminate the distinction between physical and virtual content from the perspective of the user. A text entry method for AR should thus be consistent with the broader experience and maintain any developed sense of immersion.

In response to the identified challenges, this chapter presents a novel system that enables users to type on a virtual keyboard using a head-mounted AR device and hand localisation derived from body-fixed sensors. This system is subsequently referred to as the Virtualised Input Surface for Augmented Reality (VISAR) keyboard. The VISAR keyboard is a probabilistic auto-correcting translucent keyboard system with variable occlusion, specifically designed for AR HMDs, such as the Microsoft HoloLens. The design of VISAR is underpinned by six design principles for AR text entry which are distilled from the literature and prior experience in text entry design. The system seeks to leverage learned keyboard interaction behaviour and exploit the additional dimensionality of the design space available in AR. By adapting a state-of-the-art probabilistic decoder, people are able to type in a fashion that is familiar and akin to typing on their mobile phone keyboard or on a wall-mounted touch-capable display. This is the first investigation of providing a touch-driven text entry method specifically designed for AR and based upon body-fixed (as opposed to space-fixed) sensor data. The system is thus fully encapsulated by the head-mounted device and enables truly mobile, unencumbered text entry for AR. Furthermore, the system seeks to specifically address the unique design

requirements of optical see-through head-mounted AR by accommodating design objectives that relate to the constrained display size and therefore explores minimising occlusion of the user's field-of-view.

The investigation described in this chapter is built upon four user experiments. These experiments seek to isolate the role probabilistic inference can play in AR text entry and the features that are necessary to make it effective. Experiment 1 compares the VISAR keyboard with a non-probabilistic baseline to evaluate the potential performance benefits of the error tolerant, touch driven interaction method. Encouraged by the potential of the VISAR keyboard revealed in Experiment 1, Experiments 2 and 3 examine two specific interaction features enabled by the application of probabilistic inference: i) a method for providing greater control over the decoding functionality; and ii) the ability to reduce occlusion of the user's field-of-view. After making further improvements, such as including word predictions, Experiment 4 compares the now enhanced VISAR keyboard with the similarly improved non-probabilistic baseline. The key findings of these experiments are now briefly summarised.

Experiment 1 revealed that novice users with minimal practice reach entry rates that are comparable with the current standard interaction technique used within the Microsoft HoloLens default system keyboard, which requires that users move their head to place a gaze-directed cursor on the desired key and then perform a hand gesture. However, Experiment 1 finds that in terms of discrete selection events, the virtual touch technique used in VISAR is on average 17.4% faster than the baseline method.

In Experiment 2 the effect of allowing users to seamlessly shift from probabilistic auto-correcting text entry to literal text input without an explicit mode-switch was investigated. The results revealed that users most commonly exploited the inferred-to-literal fall-back method to pre-emptively enter words they did not expect the decoder to recognise. The inferred-to-literal fall-back method introduces a speed penalty due to the requirement to dwell on a key to make a selection. Despite this penalty associated with dwell, for phrases with a high degree of uncertainty under the language model, participants were able to type as quickly as they did without the fall-back method but with reduced character error rates.

Experiment 3 revealed that the interaction techniques applied in VISAR enable users to type effectively even when key outlines and labels are hidden. Out of the 12 participants who completed both Experiment 2 and 3, 10 achieved their highest entry rates in one of the two reduced occlusion configurations examined. This shows that the majority of participants were able to readily transfer their learned typing skills to the novel interface approach. Varying keyboard occlusion in AR typing has not been proposed or explored before.

Experiment 4 returns to the baseline comparison but with the design improvements identified in Experiments 1 to 3 incorporated into VISAR and with the addition of word predictions. User

performance was evaluated under extended use with participants typing between 1.5 to 2 hours in each condition over a fixed number of test blocks. The refined VISAR design achieved a mean entry rate of 16.76 words-per-minute (wpm) compared with 14.26 wpm in the baseline condition. Analysing only the results from the final four blocks in each condition (i.e. after the most pronounced learning effect has subsided), the mean entry rates are then 17.75 wpm and 14.84 wpm for the VISAR and the baseline conditions respectively.

Finally, a validation study was conducted, which demonstrated that the VISAR keyboard is a viable text entry method for typical text entry tasks anticipated for productive use of AR HMDs. The user experience of the system is examined in four sub-tasks involving transcription, composition, replying to a message, and freely annotating real world objects.

Future text entry design for AR HMDs is informed by the results of these four experiments and the one validation study which investigate the implications of the design choices in the VISAR keyboard. To summarise, the three key novel contributions of this chapter are:

1. Six design principles informed by the literature and prior interface design experience.
2. A novel keyboard system specifically adapted to AR HMDs leveraging probabilistic inference to enable an error-tolerant touch-driven interaction paradigm.
3. Empirical results from a comparison with a gaze-then-gesture baseline entry method, and an investigation of the influence of various design decisions.

6.2 Related Work

The technological advancements in head-mounted displays are frequently accompanied by research seeking to empower their users with productive text entry methods. This section reviews the literature relevant to developing efficient text entry for HMDs. Text entry is an active field of research that has produced a plethora of input techniques. The scope in this section is constrained to work which informs the design of text entry techniques in AR, both from an interaction and language perspective.

First, efforts to support productive text entry in circumstances where user selections are subject to high levels of error are reviewed. This includes circumstances such as are potentially encountered in HMD contexts exploiting coarse hand tracking. Next, the research that specifically targets productive text entry for HMDs in both virtual and augmented reality is examined. Last, this section looks at work which explores providing mid-air text input not necessarily accompanied by a head-worn display.

6.2.1 Intelligent Text Entry

A distinction can be drawn between text entry methods which simply insert a selected letter versus those which provide intermediary intelligence to infer user intent. Patterns in language and knowledge of human behaviour can be exploited to improve text entry performance (both in terms of entry rate and error rate) for a given input technique. Adding intelligence to a text entry method is of particular value when the input process is constrained by some physical limitations.

Small keyboards constrained by the size of device on which they are deployed, are an obvious target for application of smart methods for inferring input and correcting errors. Goodman et al. [53] demonstrated the potential of a character-level language model to help reduce error rate by inferring user's intended key presses on a PDA soft keyboard. Thumb typing on small keyboards is similarly error prone due to key occlusion. Faster entry rates typically also exacerbate error rates. For example, Clarkson et al. [23] observed that participants could thumb type on a small QWERTY keyboard at 31.72 wpm in session 1 with error rates of 6.12% but that this then grew to 60.03 wpm with error rates of 8.32% after 20 sessions of 20 minutes. Kim et al. [79] evaluated a small wrist worn keyboard intended for a wearable computing environment that allowed users to achieve 18.9 wpm after five 20 minute sessions. They also noted an issue with small keyboard size and error rates which were consistently averaging above 6% over the five sessions. To address such error rates in small keyboards, Clawson et al. [24] applied a decision tree approach to correct 32.37% of user errors on a miniature physical thumb typing keyboard. Kristensson and Zhai [90] use an alternative approach based on pattern-matching to compare user's tapped points to ideal point templates. In a pilot evaluation of the entry technique on a stylus keyboard, peak entry rates in the range of 37.7 to 51.8 wpm were achieved.

Rather than relying on a character or symbolic approach to text input, it is possible to exploit people's most efficient language communication channel: voice. In situations where there is low environmental noise and privacy is not a concern, speech recognition offers the potential for very fast input. Accuracy has historically been a problem for speech-to-text input but recent advances have substantially lowered error rates [67]. Entry rates can also be improved by providing effective interfaces to support rapid correction of recognised speech. Pick et al. [145] investigated text entry in a CAVE virtual environment using speech recognition with correction taking place via a hand-held point-and-click device. Participants achieved an average of 23.6 wpm with word error rates of 0.56%. The SpeeG2 [71] interface allows text input via speech recognition with correction taking place via gestures sensed by a depth sensor. SpeeG2 supported entry at 21 wpm with users interacting in front of a wall-sized display.

In general, user performance can be considerably enhanced by exploiting the predictability of language and behaviour. There is, however, an inevitable trade off against agency and false positives should the user intent and inferred intent diverge in an intrusive fashion.

6.2.2 Text Entry for Head-Mounted Displays

Determining how best to enter text or other symbolic input in VR has been a long standing research problem. Early systems allowed short text input or annotation of the virtual environment via handwritten graphical notes/drawings (e.g. [146]), audio annotations (e.g. [181, 61]), or via hand gestures sensed with a glove (e.g. [153, 93]). Bowman et al. [14] compared four different input methods for entering text in VR while wearing an HMD: speech recognition (performed by a human), pinch gloves, a pen and tablet, and a chord keyboard. Entry rates were: 4 wpm chord keyboard, 6 wpm pinch gloves, 10 wpm pen and tablet, and 13 wpm speech.

Yu et al. [199] also evaluated three alternative text entry methods for VR by exploiting the fine head-motion tracking capability of modern HMDs. Using a head-fixed fixed gaze cursor, the three methods examined alternative approaches to indicate a key selection: dwell, button-press on a game pad, and gaze-based-gesture path (also using a game pad to indicate gesture start and finish events). In a comparative study, the three methods achieved average entry rates of 10.59, 15.58 and 19.04 wpm respectively in the sixth session (eight phrases per session). After further refinement of the gaze-based-gesture entry method, including correcting for an observed performance difference between head movement up and down versus left and right, an average entry rate of 24.73 wpm was achieved after 8 sessions of typing the same 10 phrases repeatedly.

Text entry for augmented reality has received less attention in the literature. The Augmented Reality Keyboard (ARKB) [98] uses a stereo visible light camera on an HMD to track coloured markers attached to a user's fingers. ARKB detects collisions with a virtual QWERTY keyboard displayed in the HMD. No user trial results were reported.

SwipeZone [55] exploits the touch region on the side of Google Glass to deliver a text entry method involving swiping to select key groups then the desired letter. In a controlled experiment involving entry of 10 five-letter words per block for 20 blocks, participants achieved a mean entry rate of 8.73 wpm in the final block. Also focusing on the Google Glass, Yu et al. [200] present a one-dimensional unistroke gesture technique that allows text input. The input system made use of a probabilistic stroke and language model. In their second study session, participants were able to enter words at 9 wpm.

The PalmType system [190] used Google Glass to display a QWERTY keyboard interface on a user's palm. In a user study that used a Vicon tracking system, users were able to type on

the palm of their hand at 8 wpm. Using a wrist-worn IR sensor users typed at 5 wpm. Input was literal with no auto-correction algorithm.

The various studies of text entry methods specifically targeting HMDs suggest typical entry rates in the range of 5 to 25 wpm without use of a physical keyboard. It should be noted, however, that the approaches that prove effective for VR may not transfer well to AR and vice versa. The distinction between AR and VR is not always meaningful, particularly in terms of the physical execution of interactions and more generally in aspects of software architecture. There are, however, prominent distinctions between AR and VR that should not be ignored. For example, the fact that many VR systems are tethered or for other reasons preclude extended user mobility means that the use of controllers or other input devices and fixed sensing infrastructure may be appropriate. Tracking provided by input devices or fixed infrastructure is likely to be characterised by higher accuracy and lower system delays. Similarly, the lack of control over the background scene in AR means that the visual features of a text entry interface may need to be considerably different from the same approach used in VR.

6.2.3 Mid-Air Text Entry

A variety of work has looked at how to capture input for text entry with unobtrusive sensing rather than with input-specific devices such as a glove or strap-on mini-keyboard. DigiTap [147] uses a wrist-mounted camera to detect the thumb touching 12 different points on a user's finger. No formal user study was presented, but one of the authors was able to achieve 10 wpm entering text via a literal multi-tap keyboard mapped to the finger locations.

Markussen et al. [114] evaluate three alternative text input methods for interactive large wall displays. Hand tracking was provided by an OptiTrack system and a glove with reflective markers. Hand movements were mapped onto the wall display and 'taps' were recognised based on an angle threshold of the vector between the hand and fingertip. After six 45 minute sessions, participants recorded a mean entry rate of 13.2 wpm in the best performing condition: a projected full QWERTY keyboard implementation in which users would select individual keys by moving their hand and then 'tapping'.

Iterating on these techniques, Markussen and colleagues produced Vulture [115], a word-gesture keyboard [202] designed to operate in mid-air. Again using a wall-sized display and an OptiTrack motion-capture system, users wrote at 21 wpm. As is typical with word-gesture keyboards, a probabilistic language model and template matching algorithm was used.

AirStroke [128] allows users to type using freehand gestures based on the Graffiti alphabet. The non-gesturing hand can also be employed to select word predictions. When word predictions were included, participants were able to achieve entry rates of approximately 13 wpm and with error rate under 5%.

The emergence of sensors that support fine hand and finger articulation tracking, such as the Leap Motion, has enabled the investigation of a variety of free-hand mid-air text entry techniques. Sridhar et al. [171] demonstrate text entry using a multi-finger gesture set. Using a repeated word evaluation as proposed by Bi et al. [10], participants achieved a mean peak performance of 22.25 wpm. The Air Typing Keyboard (ATK) [198] allows ten-finger typing in mid-air with the location of fingers detected by a Leap Motion depth sensor. A probabilistic decoder is used to infer the user's typing, although ATK did not model insertions or deletions. The ATK thus allows users to type as they would on a typical mechanical keyboard by extending their fingers as if pressing keys. Users were reportedly able to type at 29 wpm after an hour of practice, however, stimulus phrases were purposely selected to only include words within the vocabulary.

Fully articulated hand tracking to support 10 finger mechanical-keyboard-like text entry is perhaps the holy grail for mid-air text entry. The work of Yi et al. [198] suggests that this may be achievable but currently available sensor technologies face difficulties supporting such an interaction in a fully mobile AR use case. The requirement for on-body sensors to perform hand localisation and articulation estimation is challenged by observability constraints and independent, non-rigid movement of appendages. These issues may be remedied to some extent through careful sensor placement and improved sensor design with the AR HMD use case specifically in mind. As it stands, however, the various studies suggest that hand-based mid-air text entry, even with external fixed sensors, is typically in the range of 10 to 30 wpm. The interactions that are feasible under external tracking of hand position and articulation are not necessarily feasible in a body-fixed sensing scenario.

6.3 Approach

This chapter begins by reviewing work relevant to the design of an effective text entry method for AR HMDs. A set of six key design principles is distilled from the literature and prior experience derived from AR interface design. These design principles inform the design of the VISAR keyboard which is described in detail in Section 6.5. The results of the five experiments briefly described above are then presented. Finally, the chapter is concluded with a discussion of the limitations and open issues related to designing and deploying a fully featured AR keyboard based on a touch-driven paradigm.

6.4 Design Principles

The VISAR keyboard is intended to satisfy the design goal of providing an efficient and accurate text entry method for use in AR. The design of the VISAR keyboard is guided by six key principles that are proposed as critical features to an effective AR text entry solution. These design principles are derived with reference to the recognised features that make conventional two-dimensional text entry methods effective as well as the unique requirements of immersive interfaces for AR.

It is envisioned that the majority of mobile AR text entry use cases will be light text entry only, i.e. occasional entry of usernames, passwords, search terms or short phrases. This imagined usage behaviour also informs the following design principles.

DP 1. Rapid Input Selection

Presenting a virtual keyboard using a HMD enables a variety of potential selection methods. Text entry may be thought of as a connected sequence of discrete key selections. In order to maximise overall entry rates, interaction techniques that support rapid selection are preferable. The speed of key selection does, however, expose a potential trade off against input error, hence, the following corresponding design principle, Tolerance to Inaccurate Selection.

DP 2. Tolerance to Inaccurate Selection

Mid-air keyboard text entry by hand tracking is an inherently noisy process with a high risk of false identification of intended keys. Unmitigated, the error prone user is likely to fall back on a closed loop strategy involving selection then review. This strategy is highly detrimental to entry rate. To mitigate the inaccurate selection process, it is possible to interpret the user's input via a probabilistic decoder which treats each attempted key press by the user as an uncertain *observation*. The decoder then decodes an *observation sequence* of such key presses into individual words by assigning a posterior probability distribution over candidate words. Performance of the decoder is related to the span-length of the statistical language model and the size of the text corpora.

DP 3. Minimal Occlusion of Field-of-View

AR optical see-through displays are designed to allow users to be highly mobile and maintain visibility of the real environment. The current commercially-available devices provide fairly limited display regions that are unavoidably located in the centre of the user's field-of-view. This constrained display real estate introduces unique considerations related to the placement and styling of content. Future iterations of the AR HMDs will likely seek to expand the usable display region. While there are no doubt technical challenges that will make this difficult to achieve, AR delivered over the user's full field-of-vision is an eventuality for which HCI

researchers should prepare. Even in this eventuality, however, it will be preferable to avoid obscuring the user's central field-of-view when non-essential so that they may continue to attend the physical environment. Supporting text entry under this constraint is thus a necessary but difficult to accommodate design objective. Where possible, text entry methods should seek to minimise the occlusion of the real-world.

DP 4. Intelligent Word Predictions

When key selection speed is limited due to physical constraints on the interaction method, entry rates may be significantly increased by exploiting word predictions based on probabilistic language models. The user may only need to type several characters before the desired word is presented as a predicted option based on the prefix. Fortunately, users are increasingly exposed to such models within many mobile text entry keyboards and so the same approaches may be readily applied in AR.

DP 5. Fluid Regulation between Input Modes

The application of a decoder to auto-correct typing mistakes and errors due to sensor noise is discussed within *DP 2* above. However, sometimes users intend to write text which is unlikely to be predicted by a statistical decoder, for instance usernames or passwords, which are often intentionally designed to exhibit high perplexity under a statistical language model. It is therefore important to support *fluid regulation* of the user's uncertainty which allows users to easily indicate to the system whether they desire their key presses to be decoded or to be interpreted literally.

DP 6. Walk-up Usability and Acceptance

It is notoriously difficult to design a text entry method that will be adopted by users. Hundreds of text entry methods are proposed in the HCI literature but very few achieve mainstream adoption. A theory in economics known as *path dependence* helps explain this phenomenon [28] (for an alternative view, see [100]; see also [203, 88]). In order to use a new text entry method, users need to invest learning time. If the text entry method is radically different from a QWERTY keyboard, this learning time can be substantial.

Also relevant in the context of text entry applications for AR HMDs is the requirement to smoothly transition into other tasks. For example, users may wish to effortless transition between labelling an object with text and adjusting its position. Modes of interaction that minimise the effort required to switch between discrete tasks are thus desirable.

In summary, the above design principles guide the design decisions behind the development of the VISAR keyboard. The following section describes the system design, and where relevant, reference is made to the corresponding design principle that has informed the choices made.

6.5 VISAR System Design

The VISAR system splits function between two principal components: the decoder and the mid-air virtual keyboard. The mid-air virtual keyboard provides the visual interface and supports the direct-touch interaction technique. The decoder provides corrections of noisy key selections made on the virtual keyboard. Together they deliver a natural and immersive interaction method that enables text entry at moderate speed with acceptable error rates.

6.5.1 Decoder

Due to inaccuracies in the tracking of a user's hand and in the perceived location of the virtual keyboard, the recorded tap location and that of the user's actual intended key target will very likely differ. To infer a user's intended text from this noisy input data, the VelociTap [187] decoder was extended. Enabling error-tolerant typing is anticipated to allow users to maintain higher entry rates according to *DP 2*. Details from [187] are repeated here for completeness.

The decoder allows users to enter text by tapping out all the letters of a sentence on a touchscreen virtual keyboard. After entry, the entire sentence of noisy touch locations is provided as the input observations for decoding. The decoder searches for the most probable sentence that is consistent with the observations but that is also probable under a language model. The goal of the decoder's search is to find the sequence of actions that consumes all the observations and does so with the highest probability. The first action the decoder can take is to generate a character from the keyboard. The probability of generating a character is based on the likelihood of the observation's location under a two dimensional Gaussian centred at the key labelled with that character. The two-dimensional Gaussians are axis-aligned with two separate parameters controlling the x- and y-variances. All keys share the same two parameters. This action prefers characters near a tap's location, but generates new hypotheses for all possible keyboard characters with probability diminishing for further away keys.

The input sequence may contain an extra observation (e.g. if a user accidentally taps a key twice). The second decoder action is to delete an observation without outputting a character. A deletion penalty is assessed whenever this action is taken. The input sequence could also be missing an observation (e.g. if a tap fails to register). The third decoder action inserts an output character without advancing in the observations. This action proposes the insertion of all possible characters. Each new hypothesis pays an insertion penalty.

In the above actions, whenever a hypothesis proposes the generation of a character, including space, the hypothesis is assigned an additional penalty based on the probability of that character given the previously written text under a character n-gram language model. Whenever a space is output, the hypothesis is further penalised by a word language model based on the

previous words. The character and word language model probabilities are multiplied by either a character or a word scale factor. Words that are not in a known vocabulary list incur an additional out-of-vocabulary penalty.

The trade-off between speed and accuracy is controlled by a beam width. The decoder tracks the highest probability hypothesis seen at every point in the observation sequence. Hypotheses passing through that observation that are too improbable compared to this previous best probability are pruned. The decoder's search proceeds in parallel with multiple threads extending hypotheses. Once a hypothesis consumes all observations, it represents some possible text with a given probability. The decoder remembers the n-best finishing hypotheses for a given observation sequence.

To date, VelociTap has only been used for decoding an entire sentence of input. The design goal for the VISAR keyboard required that users be able to perform word-at-a-time entry. The decoder was extended to utilise known text context to both the left and to the right of the noisy input sequence. The left context is the previously written words for a given sentence. If no text has been written yet, the left context is the language model's sentence start pseudo-word. The left text provides context to the character and word language models during the decoder's search, i.e. it biases the search towards text that makes sense given what was previously written.

If right context is given, a hypothesis before finishing is assessed with character and word language model probabilities based on generating this right text. This makes sure the hypothesis makes sense given whatever comes after the word. For this application of the decoder, the right context was always set to be a space. This had the effect of biasing the search towards complete words.

The language models were trained using SRILM [173] on billions of words of twitter, usenet, blog, social media, movie subtitle, and web forum data. A 12-gram character model was trained using Witten-Bell smoothing. A 4-gram word model was trained using interpolated modified Kneser-Ney smoothing. Entropy pruning reduced the model size to 578 MB for the character model and 1.0 GB for the word model. The decoder incorporated a vocabulary of 64,000 words.

The free parameters of the decoder such as the variances, deletion penalty, insertion penalty, out-of-vocabulary penalty, and language model scale factors were optimised with respect to development data recorded by two of the study's researchers and two other volunteers who did not take part in any of the user studies. The development data consisted of sentences not used in the user studies. As will be discussed, experiments were performed with a normal virtual keyboard with key outlines and labels as well as keyboards with reduced visual features. Two sets of parameters were optimised for this purpose, one for the normal keyboard and one for keyboards with reduced visual features.

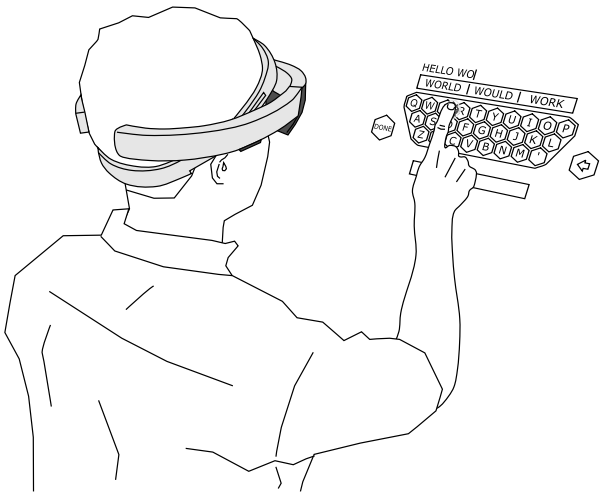


Fig. 6.1 Illustration of a user typing on the VISAR keyboard.

6.5.2 Mid-Air Virtual Keyboard

The Microsoft HoloLens provides the hardware platform for the implementation of the mid-air virtual keyboard. The HoloLens is a head-mounted see through display which also provides coarse hand-tracking. The HoloLens constructs and maintains a spatial map of the environment. This map can subsequently be exploited to maintain fine tracking of the user head location and orientation within the environment. Virtual objects can then be placed in the user's view such that they appear fixed within the local environment.

The virtual keyboard in this study is generated as a two-dimensional panel of keys. Figure 6.1 illustrates the virtual keyboard concept. The keyboard layout employed in this study was simplified to contain only characters *A* to *Z* and apostrophe (total of 27 character keys). The *SPACE* key is used as the trigger to activate the decoder on the most recent observation sequence. The *DONE* key is used in the experiments described to indicate completion of the set phrase. The *BACKSPACE* key removes previous touch input unless a space was entered and the word decoded. If the previous user action was pressing the *SPACE* key (and hence a decode), then pressing *BACKSPACE* would remove the whole previous word allowing the user to re-enter the desired word from scratch. The *SPACE*, *DONE* and *BACKSPACE* keys were always treated deterministically in that user selections immediately activated their corresponding behaviours (as opposed to being inferred). For this reason, the three control keys are distinctly separated from the 27 character keys as shown in Figure 6.1.

The experiment application runs on the HoloLens and communicates with the decoder over a dedicated wireless network. This system architecture was chosen to enable concise separation of functionality and support parallel development. There are, however, no known obstacles to performing the decoding step on-device and this will be an objective of future work.

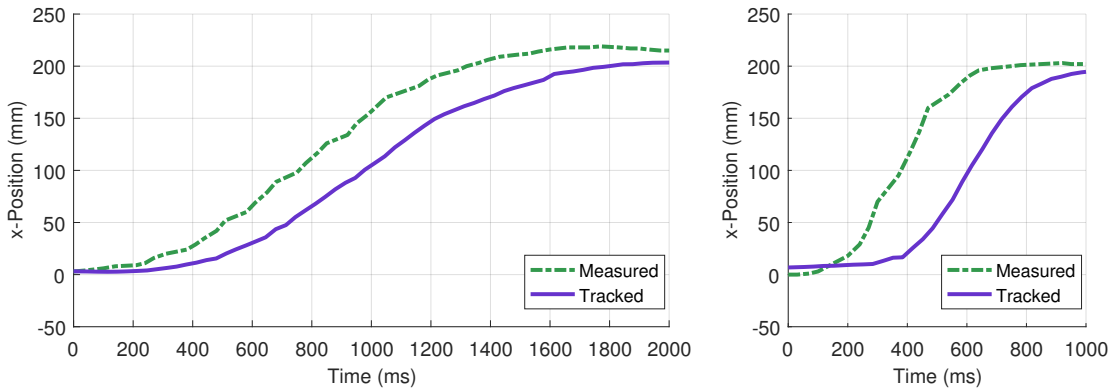


Fig. 6.2 Typical tracking delay observable in index cursor positioning as derived from the reported hand location. The step movement was generated by moving the hand approximately 200 mm along a single axis. Measured position was approximated from the on-device video recording. Tracked position was logged and synchronised with the video. Left plot shows a slow hand movement lasting approximately 2 s. Right plot shows a fast hand movement lasting approximately 1 s.

6.5.3 Virtualised Touch Key Selection

The VISAR keyboard seeks to minimise learning time (*DP 6* in Section 6.4) by exploiting an interaction technique that is compatible with people's pre-existing keyboard typing skills and experience. The HoloLens provides access to the hand position, as tracked by the on-device sensors. The documentation does not explicitly define the point being returned as the tracked hand position but visual inspection suggest that it approximates the centre of the dorsum of the hand, i.e. surface opposite the palm. No hand orientation information is available. The tracked hand position is exploited to place a cursor approximating the tip of the index finger. It is important to understand that the index finger is not tracked and so this cursor placement is only approximate. The cursor remains at a fixed offset and orientation with respect to the tracked hand position, and does not adjust for joint articulation or hand orientation changes.

The tracked hand position visibly lags behind the true hand position. The typical lag in hand position tracking (approximately 220 ms) as experienced by the user in this study is shown in Figure 6.2. Nevertheless, the reported tracked hand position shows good robustness to pose and joint articulation changes. The lag also appears to be largely constant, allowing the user to accommodate the delay in their pointing behaviour. Although testing an indirect control task, Hoffmann [68] suggests that the transition between a continuous and a 'move-and-wait' control strategy occurs at around 700 ms. Chung et al. [22] evaluate actual hand movements under time delay in a virtual environment. They find that below 440 ms, the target width dominates

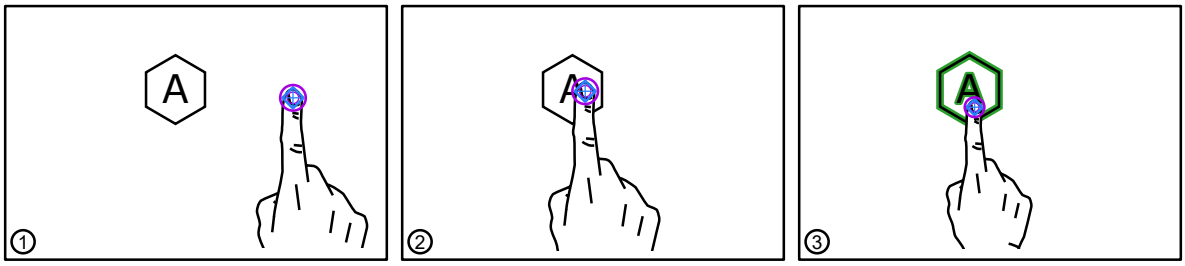


Fig. 6.3 Virtualised touch driven key selection sequence. 1) The hand is in the ready position with the index cursor showing. 2) The user moves their hand to the desired key location. 3) The user moves their hand forward towards the key to generate an intersection between the key and the index cursor. The key is selected.

the movement amplitude in determining movement time under delay. The application of the decoder to mediate touch inaccuracies can be thought of as increasing the effective target width of a key and thus also acts to reduce the detrimental effects of the tracking delay. More generally, provided tracking delay remains within the range associated with a continuous control strategy, the effect of delay can largely be reduced by ensuring key sizes are of a reasonable size.

It is important to highlight that the hand tracking behaviour resulting from the various device limitations is not ideal. An ideal system for single-finger typing would provide robust index finger position and articulation tracking with minimal delay. Superior multi-finger tracking might also enable ten finger two-handed typing. As the enabling tracking technology develops for integration with head-mounted AR, such enhancements to the system presented here are worthy of examination. Nevertheless, even more advanced technologies are unlikely to deliver perfect tracking in the fully mobile use case with no off-body sensors. There is thus likely to be ongoing demand for intelligent mediation of mid-air touch based interactions. The approach and design presented in this chapter for touch-driven interaction facilitated by intelligent decoding is sufficiently flexible and extensible to accommodate advancements in the underlying tracking technology. While the current iteration is subject to several key limitations, it nevertheless provides a valuable baseline and foundation for future development.

To perform direct touches on the keyboard, the user moves their hand to generate an intersection between the keyboard plane and the cursor approximating the tip of their index finger. Upon intersection, the user's touch location is indicated by a small circular marker while the nearest key flashes to green then fades back to white. The intersection point is added to the trace point list and the nearest key is added as a key press. This key selection approach is illustrated in Figure 6.3.

6.5.4 Experimentally-Driven Design Iteration

The sequence of four experiments reflects several key design iterations of the VISAR keyboard. Each experiment is described in detail in the remainder of this chapter but as an aid to the reader, the design journey taken with the VISAR keyboard is briefly outlined. The experiments conducted are typical of user studies in text entry with participants recruited to perform a standard text transcription task. Participants are instructed to copy stimulus phrases as quickly and accurately as possible in the various test conditions. The experiments therefore provide results indicative of potential user performance and experience.

Experiment 1 (see Section 6.6) evaluates the virtualised touch driven approach against an existing text entry method derived from the standard HoloLens system keyboard. This experiment principally seeks to investigate walk-up usability and acceptance (*DP 6* in Section 6.4) in comparison with an established baseline.

The results of Experiment 1 demonstrate only marginal differences in net entry rate but the time between key selections using the VISAR keyboard is significantly faster. This observation is in alignment with *DP 1* in Section 6.4 but the failure of more rapid key selections to translate into faster entry rates highlighted a potential design flaw. It was hypothesised that this flaw related to the frequency of error correction undertaken by users. One interesting source of errors, and a unique consequence of decoder based touch mediation, is the inability to distinguish between unknown or unusual words and an erroneous input. As an example, one participant correctly input the key sequence 'D-Y-N-E-G-Y' corresponding to the company name 'DYNEGY' but was overridden in the decode step to the word 'SUNG'. This observation, among others, suggested the investigation of methods for allowing users to indicate a literal interpretation of their input sequence.

In broad terms, the user can provide such guidance to the decoder according to two alternative strategies: proactively or reactively. Fortunately, the use of one strategy does not preclude the other and both can be combined to provide multiple correction pathways for the user. A proactive approach may involve the provision of some additional information during input that the decoder can exploit to better distinguish intent. In Experiment 2, a proactive approach to literal input disambiguation was specifically explored. The implementation takes significant inspiration from Weir et al. [192] who augment touchscreen text input by incorporating touch pressure as an indicator of the degree of confidence associated with an input event. Instead of pressure, a depth based transition is used to switch fluidly between input modes in alignment with *DP 5* in Section 6.4.

In contrast, a reactive approach is used in many conventional mobile keyboards and implemented as a set of several alternative word predictions, one of which may be the as-input literal word. The user may then either opt-out of replacing their literal input with one of

the predictions or choose the literal input from among the displayed options. Note that this strategy is actually introduced as part of the addition of word predictions to VISAR and tested in Experiment 4. The results of Experiment 2 show that the proactive strategy employed did not substantially impact the performance of the keyboard but was rated as useful by a majority of participants.

It is not uncommon for users to type with a single finger on their smartphone, tablet or even on large interactive information displays such as are found in shopping centres. Broad familiarity with single finger typing gave rise to the hypothesis that users may be able to exploit this prior experience, and possibly any ingrained muscle memory, to touch type on the virtual keyboard with reduced visual features. This commonality with the physical interaction paradigm also promotes walk-up usability and acceptance (*DP 6* in Section 6.4). The exploration of reduced visual features specifically is motivated by the objective of minimising the occlusion of the physical world by virtual content (*DP 3* in Section 6.4). In Experiment 3, it was found that the majority of participants were in fact able to type effectively even when all key outlines and key labels were removed.

Encouraged by the faster inter-key timings observed in Experiment 1, the flexible design space demonstrated in Experiment 2, and the performance achieved with the minimal occlusion keyboard in Experiment 3, further design refinements were applied to the VISAR system. Most significantly, word predictions were added as per *DP 4* in Section 6.4. Performance was re-evaluated against a similarly revised baseline in Experiment 4. As briefly described above, the implementation of word predictions also supported reactive disambiguation of intent by presenting the literal input as one of the alternative word panels. This functionality is an alternative but complementary reflection of *DP 5* in Section 6.4. These various design improvements result in higher entry rates and lower error rates with one participant achieving a peak mean typing speed of 23.38 wpm in an experimental block.

Finally, the VISAR keyboard is demonstrated in a range of envisaged short text entry tasks conducted by participants while standing and independently moving through a physical environment. Participants achieved tolerable entry rates in these conditions despite very little prior training and indicated good acceptance of the system design.

6.6 Experiment 1: Selection Method Evaluation

Experiment 1 examines the hypothesis that allowing users to engage with the keyboard through direct touch is more intuitive and will deliver higher text entry rates than a gaze-then-gesture interaction. The gaze-then-gesture interaction method is the primary selection paradigm exploited on the HoloLens.

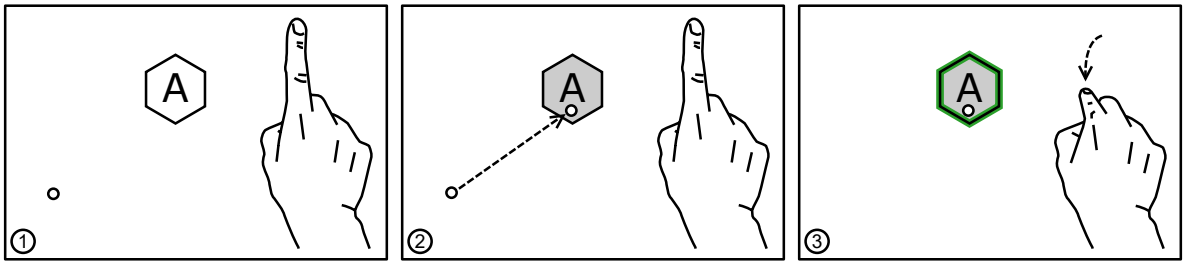


Fig. 6.4 Gaze-then-gesture key selection sequence. 1) The user is not looking at the key and the gaze cursor is in bottom left of frame. 2) The user looks at the desired key and the key highlights when the gaze cursor enters its region. 3) The user performs the *air-tap* gesture and the key is selected.

The gaze-then-gesture interaction method leverages the *gaze cursor* for pointing. Strictly, the *gaze cursor* does not reflect the user's eye movements but rather the orientation of the head-fixed frame. The *gaze cursor* is placed at the first intersection of the ray emanating from the head-fixed frame along the principle forward axis. For conciseness and consistency with the Microsoft documentation, the term *gaze* is used in this chapter to refer to this head-tracked vector at the cost of semantic accuracy. The *gaze cursor* is thus the point of intersection of this vector with objects in the scene.

The gaze-then-gesture interaction method is based on designating focus with the gaze cursor and making a selection using the *air-tap* gesture. The *air-tap* gesture involves placing the hand first in the neutral position with only the index finger raised. The index finger is subsequently pressed down then raised again to indicate a selection. The gaze-then-gesture paradigm is used in the system keyboard provided by default in HoloLens applications. To specify a key, the user focuses the gaze cursor on the desired letter then performs an air-tap gesture to make the selection. The visual appearance of the key in focus changes to provide feedback on which letter will be selected when the selection gesture is performed. The gaze-then-gesture interaction sequence is illustrated in Figure 6.4.

The gaze-then-gesture based keyboard evaluated in this experiment serves as an established baseline method against which the VISAR system is compared. To provide a valid reference point, an experience is delivered that replicates use of the HoloLens system keyboard while at the same time standardises certain design features for the sake of a meaningful comparison. In particular, an express choice was made not to integrate the decoder with the gaze-then-gesture based keyboard as the underlying paradigm implies a discrete and two-step process of selection then confirmation.

6.6.1 Method

Experiment 1 is a within-subject experiment comparing two conditions:

Fig. 6.5 Appearance of the BASELINE keyboard condition as viewed in the HoloLens.

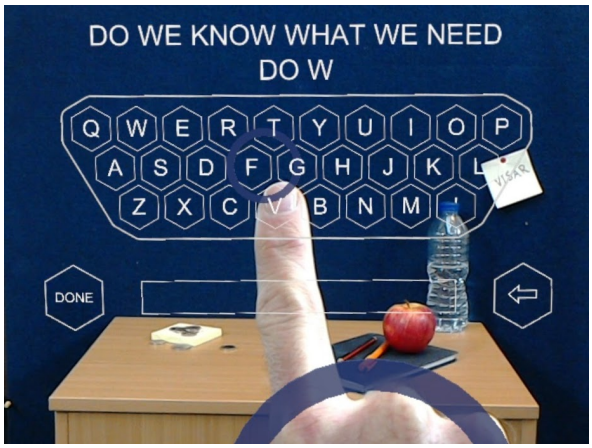


Fig. 6.6 Appearance of the VISAR keyboard condition as viewed in the HoloLens.

- **BASELINE**: Gaze-then-gesture interaction in which user moves the gaze cursor to the desired key, then performs the *air-tap* gesture with their index finger to make the selection.
- **VISAR**: Error-tolerant mid-air touch keyboard in which the user moves their hand to generate an intersection between the index finger cursor and the virtual keyboard plane to type a key. After the entry of each word, the *SPACE* key activates the decode method to replace the literal entry with the most probable word.

The appearance of the BASELINE and VISAR keyboards as viewed through the HoloLens is shown in Figures 6.5 and 6.6 respectively. Note that for all experiments reported in this chapter, the potentially confounding variable of background scene colouration and clutter was controlled for by seating participants in front of a flat colour poster board. The default sizing and distance of the system keyboard is approximately replicated for the BASELINE condition: the keyboard is placed at a distance of 1.2 m and the apparent key diameter was set to approximately 45 mm. The VISAR keyboard was placed within reaching distance (approximately 0.5 m) and scaled down to fit within the display such that the apparent key diameter was approximately 22 mm.

As described in Section 6.5.2, selecting the *BACKSPACE* key on the VISAR keyboard would either remove the previous character or the previous word depending on whether a decode step was previously triggered. Given that no decodes were performed in the BASELINE condition, the *BACKSPACE* key would always just remove the previous input key.

12 participants were recruited for a single one-hour session (4 female, 8 male). Participants received a £5 Amazon voucher in compensation for their time. Participants were briefed on the experimental protocol and then fitted with the AR headset. The order of the two conditions was counterbalanced.

Participants were instructed to type provided phrases as quickly as possible while maintaining low error rates. To compute entry rate in words-per-minute (wpm), entry time was measured from the first key press until the selection of the *DONE* key to submit the sentence. The numerator, i.e. the effective word count, is calculated based on the entered phrase length minus one (since entry time starts at first key press) divided by a nominal word length of five characters. Error rate was measured using character error rate (CER). CER is the minimum number of character-level insertions, deletions and substitutions required to transform the response text into the stimulus text, divided by the number of characters in the stimulus text. After selecting the *DONE* key to submit, participants would see a brief dialog showing their entry rate (wpm) and their character error rate (CER) for the phrase just entered. Participants were instructed that more care should be taken if their reported error rate was consistently above 10%.

The stimulus sentences were taken from the memorable phrases subset of the Enron mobile message dataset [185]. The 200 sentences in the memorable phrases subset were filtered to those with 40 or fewer character, 4 words or more, and that contained only the letters *A* to *Z* and apostrophe. The character limit was imposed to ensure that all phrases would appear on a single line and within the visible display region at the nominal keyboard placement location. The word limit was imposed to ensure a base length and complexity in stimulus phrases. The letter constraints were necessary to ensure all phrases could be typed using the simplified keyboard layout. All sentence terminating punctuation was removed. The resulting set used in Experiment 1 contained 90 distinct phrases. As described in Section 6.5.1, the decoder incorporated a vocabulary of 64,000 words. The out of vocabulary percentage for the Experiment 1 phrase set was 0.56%.

At the start of each condition, the participant was instructed to type five practice sentences. During this time, they were encouraged to ask questions and make sure they understood the interaction mechanism and keyboard functionality. Upon completing the practice phase, users began typing sentences taken at random from the stimulus set. The test phase would run for 15 minutes of cumulative entry time (sum of time between first key press on a new phrase to selection of the *DONE* key).

6.6.2 Results

Results reported contain only the phrases entered during the 15 minute test period (the initial five practice sentences are excluded from the analysis). Unless otherwise specified, statistical significance tests were performed using within-subjects repeated measures analysis of variance at an initial significance level of $\alpha = 0.05$ for entry rate and using Friedman's test for error

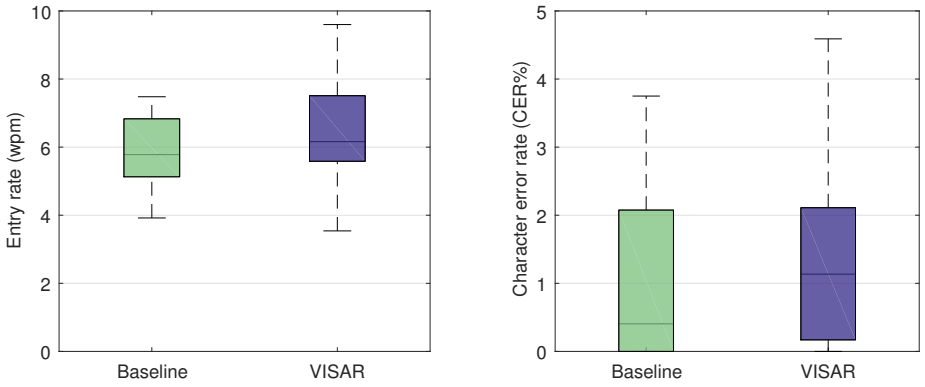


Fig. 6.7 Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 1. Entry and error rates are marginally higher in the VISAR condition.

rate (since errors are count data). Adjustments were made for multiple comparisons using Holm-Bonferroni correction.

The group descriptive statistics in each condition are shown in Table 6.1. Figure 6.7 presents boxplots of both entry rate and character error rate. The mean entry rate in VISAR is faster than BASELINE (6.45 vs. 5.86 wpm), however, the result is not significant ($F_{1,11} = 2.160$, $\eta_p^2 = 0.164$, $p = 0.170$). Participants achieved acceptable error rates in both conditions, although BASELINE yielded marginally higher accuracy. This difference was not statistically significant ($\chi^2(1) = 1.600$, $p = 0.206$).

Although the net performance difference between the two conditions is not significant, it is useful to examine the learning effect associated with each interaction technique. A text entry method that is intuitive and easy to gain proficiency in is more likely to gain traction as highlighted by *DP 6* in Section 6.4. Figure 6.8 shows the boxplots of participant entry rate corresponding to the beginning, middle and end of the 15 minute test block. Interestingly, the boxplots illustrate a more marked improvement in entry rate between the first and last interval in the VISAR condition compared with the BASELINE. The increase in mean entry rate between the first and last intervals is 19.7% for VISAR versus 8.1% for the BASELINE.

Table 6.1 Entry rate (wpm) and character error rate (CER%) descriptive statistics from Experiment 1. Results show mean $\pm$ 1 standard deviation [min, max] for entry rate and median [min, max] for character error rate.

Condition	*Entry Rate (wpm)*	*Error Rate (CER)*
BASELINE	5.86 $\pm$ 1.12 [3.92, 7.48]	0.41 [0.00, 3.75]
VISAR	6.45 $\pm$ 1.83 [3.54, 9.60]	1.14 [0.00, 4.59]

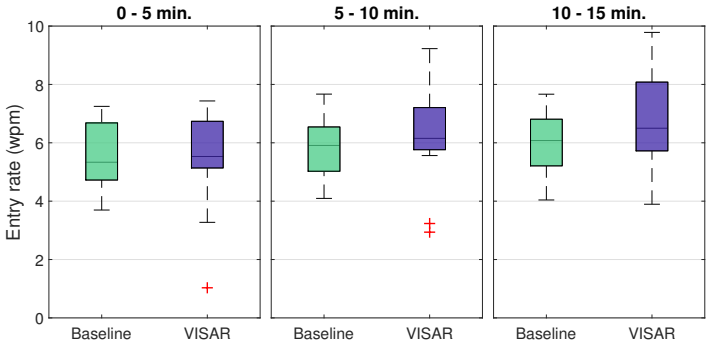


Fig. 6.8 Boxplots of entry rate (wpm) over time intervals 0-5 minutes, 5-10 minutes and 10-15 minutes in Experiment 1. Red cross indicates outlier based on $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$.

Also observable in Figures 6.7 and 6.8, however, is a larger variance in participant entry rate compared with the BASELINE. This suggests that individual user characteristics may have a more prominent influence on performance due to certain attributes of the VISAR keyboard design.

At the completion of both conditions, participants responded to three statements in a short questionnaire targeting their experience with the two conditions. The three statements are included in Table 6.2 and examined perceptions of typing speed, accuracy and comfort. Responses were recorded on a Likert scale from 1-strongly disagree to 5-strongly agree. The full response distribution is shown in Figure 6.9 while median scores are presented in Table 6.2.

The condition effect on these responses is examined using a Wilcoxon signed rank test. The participant median perception of typing speed was significantly higher in the VISAR condition ($Z = -2.365$, $p = 0.018$). This perception is consistent with actual performance in that entry rates were on average higher in the VISAR condition though not significantly so.

The participant median perception of accuracy was significantly higher in the BASELINE condition ($Z = 2.889$, $p = 0.004$). This perception is also consistent with actual performance in that lower (though not significantly lower) mean error rates were observed in the BASELINE condition. There is no significant difference in participant perception of comfort although several participants did comment on some shoulder discomfort in the VISAR condition.

Despite the marginal difference observed in raw entry rate, it was observed during the experiment that the rate of key presses appeared faster in the VISAR condition. This suggested an analysis of inter-key timing, that is, the time between discrete key selections. The mean inter-key timing per participant was computed based on the inter-key times across all test phrases, including presses of control keys and subsequently deleted letters. Inter-key timing

Table 6.2 Median questionnaire response to questions 1 to 3 in Experiment 1. Responses were recorded on a five point Likert scale from 1-strongly disagree to 5-strongly agree.

Statement		BASELINE	VISAR
Q1	The keyboard made it easy to type quickly.	2.5	4.0
Q2	The keyboard made it easy to type accurately.	4.0	2.0
Q3	The keyboard was comfortable to use.	3.5	3.0

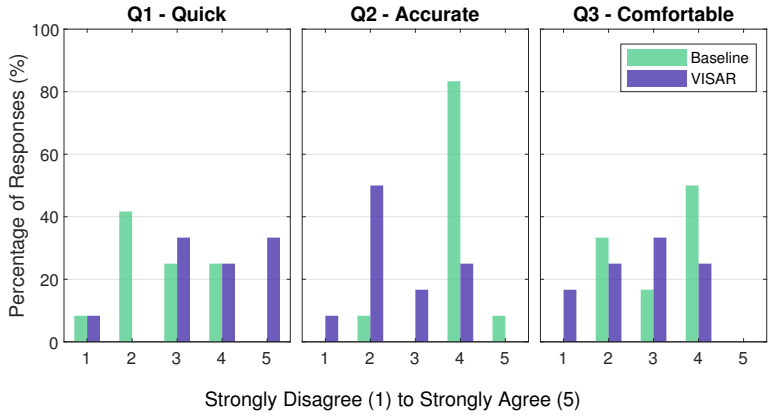


Fig. 6.9 Distribution of participant responses to Experiment 1 questionnaire. The question statements Q1-3 are defined in Table 6.2.

serves as a proxy for the upper-bound entry rate potential of the interaction method, independent of error rate.

Table 6.3 summarises the group inter-key timing results. The inter-key timing was 17.4% faster in VISAR and this difference was statistically significant ($F_{1,11} = 7.600$, $\eta_p^2 = 0.409$, $p < 0.05$). This result suggests that, while the VISAR text entry method supports more rapid key selection, it loses significant speed due to the frequency of re-corrections required due to incorrect decoder results. Recall that pressing backspace in the VISAR condition after the decoder prediction was returned would result in the whole word being deleted. This was a simple solution intended to allow users to quickly retype mistaken or incorrectly decoded input.

Table 6.3 Inter-key timing (s) descriptive statistics from Experiment 1. Results show mean $\pm$ 1 standard deviation [min, max].

Condition	*Inter-key Timing (s)*
BASELINE	2.01 ± 0.40 [1.57, 2.86]
VISAR	1.66 ± 0.47 [1.06, 2.74]

Table 6.4 Entry rate (wpm) descriptive statistics based on phrases not requiring whole-word deletions in Experiment 1. Results show mean ± 1 standard deviation [min, max].

Condition	*Minor Revision Entry Rate (wpm)*
VISAR	7.13 ± 2.24 [3.54, 10.63]

Clearly such actions introduce a corresponding performance penalty. The mean frequency of whole word deletions across participants in Experiment 1 was 7.8 and the median was 6. Assuming an error free entry rate of 7 wpm, 6 whole word deletions is approximately equivalent to a time penalty of 50s, or roughly 5% of the total experiment duration. While improvements to the error correction procedure are certainly necessary, the inter-key timing result is promising in terms of *DP 2* as described in Section 6.4 which suggests that rapid selection may ultimately help realise faster entry rates.

While re-corrections are an unavoidable reality in any recognition-based approach, it is worth evaluating the upper-bound potential for the VISAR method that might be achieved through decoder and/or interface improvements such as the provision of a literal entry fall-back method. To evaluate this potential, the mean entry rate was recomputed for participants after removing entries in which one or more whole-word deletions occurred. This analysis can provide some insight on the upper-bound potential of the method among the novice participant group. Table 6.4 shows the results based on this post-analysis. The difference in entry rate between all entries using VISAR and only entries without whole-word deletions is statistically significant ($F_{1,11} = 10.006, \eta_p^2 = 0.476, p < 0.01$). There is no parallel to a whole word deletion in the BASELINE condition and therefore it is not included in Table 6.4.

6.7 Experiment 2: Fluid Fall-Back to Precise Key Selection

Experiment 1 showed that users on average typed individual keys faster using VISAR but the overall text entry rate was not significantly improved due to the need to correct errors. Experiment 2 investigated how to mitigate this problem by allowing users to seamlessly combine inferred and literal keyboard entry. This was achieved by providing users with an optional fall-back method for precisely selecting keys as informed by *DP 5* in Section 6.4. This *precision selection mode* can be optionally activated by users and does not interfere with the normal direct touch interaction of VISAR. Letters entered using this mode are not subject to change during the decode step.

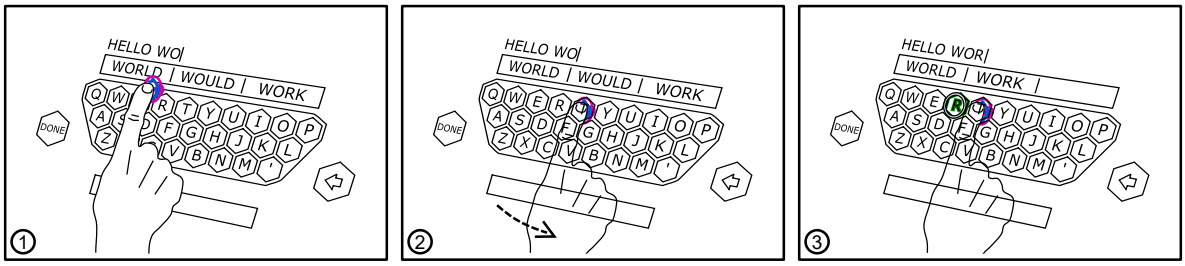


Fig. 6.10 Precision key selection sequence. 1) Hand is in ready position with index cursor showing. 2) User moves hand so that index cursor is inside the keyboard plane. 3) After 1 second, a secondary cursor appears attached to the keyboard plane. The user moves their hand to place the secondary ring cursor over the desired key. The key is selected based on a 1 second dwell period.

6.7.1 Implementing Precise Key Selection

To activate the precision key selection mode, the user pushes the index finger cursor into the keyboard and holds it on the far side of the keyboard plane. After one second, a new ring cursor appears, attached to the keyboard plane. The user can make fine adjustments to move the ring cursor so that it highlights their desired letter. The ring cursor is held on the desired key for a further one second period until the selection is confirmed by an *accept* tone. If only a single key selection is desired, the user can then retract their hand such that the index finger cursor exits from behind the keyboard plane and the interaction method returns to standard discrete touches. If multiple key selections are desired, subsequent letters can be chained together by dragging the ring cursor to a new key. This allows the user to completely specify a series of letters or an entire word without having to reactivate the precision selection mode.

The precision selection mode applies only to the letters *A* to *Z* and apostrophe. Making a precise key selection using this feature informs the word correction decoder that the specified letter cannot be changed or deleted, that is, the individual letter key selection has 100% certainty. During the briefing of the experiment, this functionality was demonstrated to participants via a video. Participants were also informed that they could specify any or all letters in a word in this manner. The precise key selection interaction process is illustrated in Figure 6.10.

6.7.2 Method

A further 12 participants were recruited for a single two-hour session (2 female, 10 male). None of the participants had taken part in Experiment 1. Note that the session time was split between execution of Experiment 2 and Experiment 3 (described later in Section 6.8). Participants

always carried out Experiment 2 before Experiment 3. Participants were compensated with a £20 Amazon voucher for their time.

Experiment 2 examined the impact of providing a precise key selection fall-back method on text entry performance. This was a within-subject design with two conditions:

- **VISAR WITHOUT FALL-BACK OPTION**: Participants touched keys to type out the phrase. The word correction decode was triggered when the participant touched the *SPACE* key. There was no provision for specifying that particular letters should be unchanged by the correction step.

- **VISAR WITH FALL-BACK OPTION**: Identical to the previous condition but with addition of the optional precise key selection mode.

The decoder's parameters were re-tuned prior to Experiment 2 based on the trace logs captured as part of Experiment 1. A new distinct set of 115 stimulus phrases was extracted from the Enron dataset for Experiments 2 and 3. Again all phrases were constrained to be 40 characters or less, four words or more, and containing only the letters *A* to *Z* plus apostrophe. The out of vocabulary percentage for the phrase set in Experiments 2 and 3 was 0.54%.

The order of the two conditions was counterbalanced. Participants began each condition by entering five practice sentences. After completing the practice phase, the test phase began and participants were presented with stimulus phrases in random order. The test phase ran for 15 minutes of cumulative entry time. Participants were encouraged to take a five minute break before moving on to the next condition.

6.7.3 Results

Participant entry rates (wpm) are summarised in Table 6.5 and Figure 6.11. The precise key selection fall-back modality was used at least once by 11 out of the 12 participants. The difference in entry rate between the with and without conditions was negligible and not significant ($F_{1,11} = 0.379$, $\eta_p^2 = 0.033$, $p = 0.551$). The very low effect size suggests that the provision of the precision fall-back functionality is unlikely to have significantly influenced text entry performance.

Table 6.5 also shows the character error rate (CER%) descriptive statistics. The median character error rate is approximately 40% less in the VISAR WITH FALL-BACK OPTION condition, however, this difference is not significant based on a Friedman's test ($\chi^2(1) = 3.000$, $p = 0.083$). The fall-back method was used on average 5.1 (median = 2.5) times on distinct words by participants. Out of those usages, 75.4% were pre-emptive in that there was no prior

Table 6.5 Entry rate (wpm) and character error rate (CER%) descriptive statistics from Experiment 2. Results show mean ± 1 standard deviation [min, max] for entry rate and median [min, max] for character error rate.

Condition	*Entry Rate (wpm)*	*Error Rate (CER)*
VISAR WITHOUT FALL-BACK	8.67 ± 1.05 [6.45, 10.09]	1.46 [0.21, 6.42]
VISAR WITH FALL-BACK	8.34 ± 1.74 [6.43, 11.13]	0.88 [0.00, 4.40]

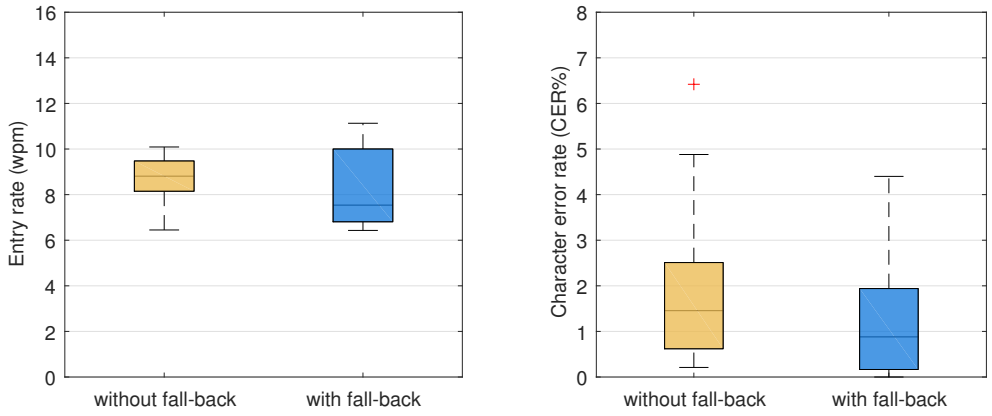


Fig. 6.11 Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 2. Red cross indicates outlier based on $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$.

word correction failure to prompt the need for precision input. In other words, participant's most often employed the precision fall-back method when they expected the decoder to fail to return the correct word.

Also worthy of investigation is whether the performance of the fall-back method was affected by the average uncertainty (information entropy) of a phrase. It is reasonable to conjecture that the fall-back method is likely to be more useful for phrases with higher self-information since these are harder to predict by the decoder. The self-information, I, expressed in bits, of a phrase is determined by computing

$$I = \log_2(1/P), \tag{6.1}$$

where P is the probability of the phrase under the decoder's character language model.

Figure 6.12 shows the usage profile of the precision fall-back method according to the self-information of the stimulus phrase set. It lists the percentage of phrases that fall within a given self-information interval for: bar 1) all phrases in the Experiment 2/3 phrase set, bar 2)

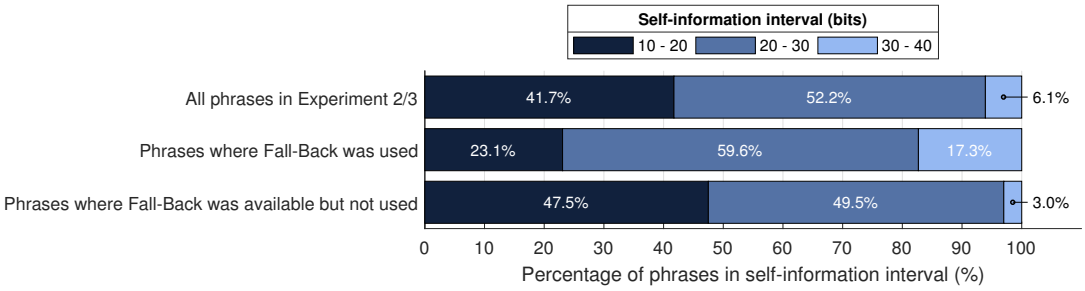


Fig. 6.12 Precision fall-back usage profile according to phrase self-information interval. Each horizontal bar shows the distribution of stimulus phrases split into three intervals of self-information. A phrase with higher self-information indicates greater uncertainty under the character language model. From top to bottom, the bars show the proportion of phrases in each self-information interval after conditioning on: 1) all phrases in the Experiment 2/3 phrase set; 2) phrases where the fall-back method was used; and 3) phrases where the fall-back method was available but not used. The middle bar reveals that the fall-back method was used more frequently when phrases of moderate to high self-information were encountered.

phrases where the fall-back method was used, and bar 3) phrases where the fall-back method was available but not used.

The usage profile indicates that the fall-back method was used more frequently in the mid and high self-information phrase intervals and less frequently in the low self-information phrase interval. This is an intuitive result given the intended purpose of providing a precision fall-back method is to assist in typing unusual or out-of-vocabulary words (i.e. words that would increase the self-information of a phrase) that might otherwise be falsely corrected by the decoder.

Figure 6.13 shows the entry rate and character error rate for the VISAR WITHOUT FALL-BACK OPTION condition and the subset of the VISAR WITH FALL-BACK OPTION condition where the fall-back method was actually used. The phrase sets are binned according to the three intervals of phrase self-information and the average entry rate and character rate are computed. In Figure 6.13, it can be seen that the entry rate is distinctly lower than that observed in the VISAR WITHOUT FALL-BACK OPTION condition when the fall-back method is used in the low (10-20) and mid (20-30) phrase self-information intervals. This can be explained by the inherent time penalty introduced by the dwell period required to activate and select in the precision fall-back mode. However, the entry rate is only marginally slower in the VISAR WITHOUT FALL-BACK OPTION condition subset for phrases in the high (30-40) self-information interval. This result suggests that the fall-back method can assist in maintaining entry rates when difficult phrases are encountered.

The character error rates shown in Figure 6.13 indicate that the precision fall-back method was not always effective at reducing errors. Indeed, it was observed that some participants

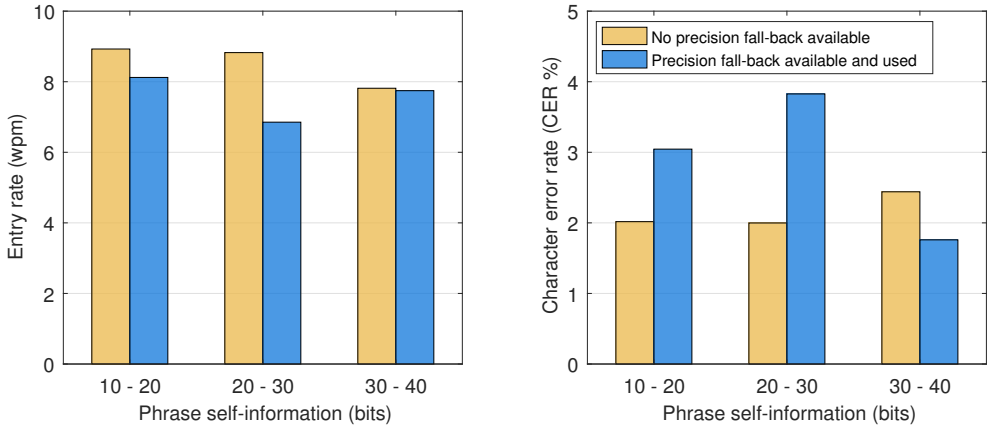


Fig. 6.13 Text entry performance in the VISAR WITHOUT FALL-BACK OPTION condition compared against the subset of phrases typed in the VISAR WITH FALL-BACK OPTION condition where the fall-back method was actually used. Entry rate (wpm) (left) and character error rate (CER%) (right) for phrases binned according to self-information.

would mistakenly add additional characters using the precision fall-back method, and (by design) these would then not be corrected by the decoder. Figure 6.13 does, however, suggest that the fall-back method was effective at reducing errors in the high self-information interval (30 - 40). The results presented in Figure 6.13 highlight that further refinement is required to ensure the fall-back method can be reliably leveraged by users. Additional training and practice in use of the method is also likely to improve performance. Nevertheless, the performance observed in the high phrase self-information interval does suggest that the precision fall-back method can help to reduce error rates without a significant cost to entry rates.

Participant subjective feedback was obtained in the form of a summative questionnaire which gauged overall impressions of the keyboard and queried specific distinctions where relevant. This short questionnaire was completed after participants finished both Experiment 2 and 3. The statements relevant to Experiment 2 are summarised in Table 6.6 along with the participant median response. The implementations of VISAR WITH and WITHOUT FALL-BACK differed only in the provision of the fall-back functionality. For this reason, it was considered reasonable to examine participant's overall impressions of using the keyboard (Q1-3) and then specifically target their experience of the fall-back functionality (Q4). Responses were recorded on a Likert scale from 1-strongly disagree to 5-strongly agree. The full distribution of responses is shown in Figure 6.14.

The responses to Q1-3 provided limited information in an absolute sense due to the lack of an alternative condition to compare against. The response to statement Q4 indicates that participants generally found the fall-back method to be useful (58.3% indicated agree or strongly agree). Only two (16.7%) participants specifically felt that it was not useful.

Table 6.6 Median questionnaire response in Experiment 2 on a five point Likert scale from 1-strongly disagree to 5-strongly agree.

Statement		*Median Response*
Q1	The keyboard made it easy to type quickly.	3.0
Q2	The keyboard made it easy to type accurately.	3.5
Q3	The keyboard was comfortable to use.	3.0
Q4	The precision selection method was useful.	4.0

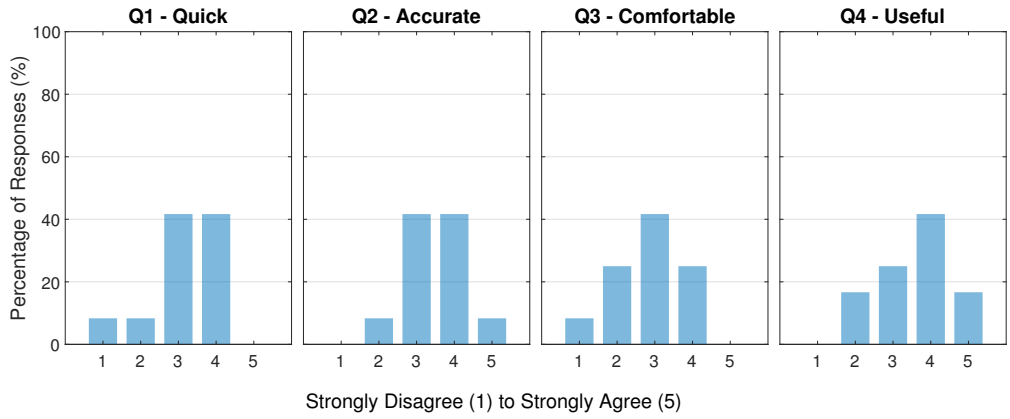


Fig. 6.14 Distribution of responses to Experiment 2 questionnaire. The question statements Q1-4 are defined in Table 6.6.

In summary, the provision of the fall-back mechanism did not deliver the increase in entry rate that was anticipated. Clearly there are other aspects to the task of dealing with unusual vocabulary and error corrections that require further investigation. Nevertheless, the negative impact on entry rate is negligible ($<4\%$) and the questionnaire results indicate that it was considered useful by the majority of participants. This suggests that the precision fall-back approach delivers some valuable functionality and may ultimately serve as a useful component within a more complete suite of error correction interactions. In conclusion, the fall-back method does not adversely affect entry rates and can help to reduce error rates when employed effectively.

6.8 Experiment 3: Minimising Keyboard Occlusion

Experiment 3 investigates the implications of design principle *DP 3* which is critical for AR HMD text entry—minimising field-of-view occlusion. By reducing the number of visual features of the keyboard which are displayed in the HMD, the user can focus their attention on

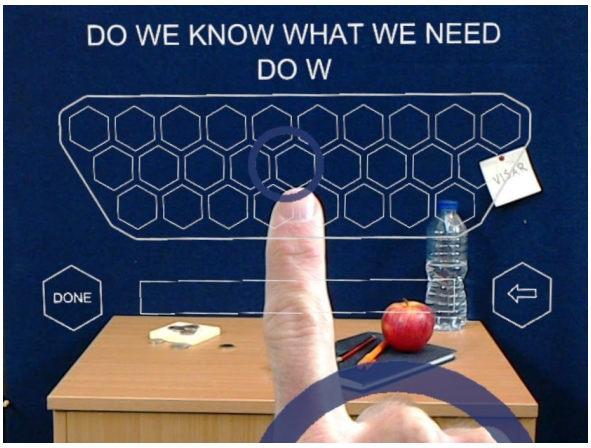


Fig. 6.15 The VISAR REDUCED OCCLUSION condition with no key labels.

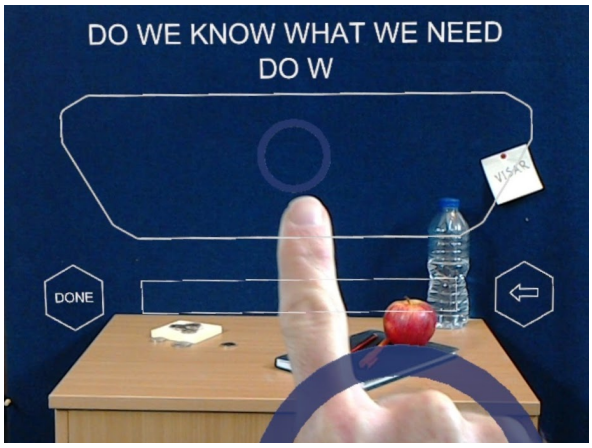


Fig. 6.16 The VISAR MINIMAL OCCLUSION condition with no key labels or outlines.

the AR scene, rather than on the text entry interface. It was hypothesised that users could be trained to type using the VISAR keyboard by just providing an outline of the keyboard and hiding both the outlines of the individual letter keys and the labels on the individual keys.

6.8.1 Method

As described in Section 6.7.2, Experiment 3 was conducted with the same participant group as Experiment 2 but in the second half of the participant's two hour session. The same phrase set covered both experiments, with stimulus phrases randomly presented without replacement. To manage fatigue, participants were encouraged to take a five minute break between conditions.

Experiment 3 was a within-subject design with two conditions:

- **VISAR REDUCED OCCLUSION**: Letter labels were removed from keys as shown in Figure 6.15. All other keyboard features remained the same. The optional precision fall-back method was also available.

- **VISAR MINIMAL OCCLUSION**: Identical to the previous condition but with the hexagonal key outlines also removed as shown in Figure 6.16.

The order of conditions was fixed to deliberately exploit the learning effect associated with a sequential reduction of visual features, i.e. participants would perform the test in the VISAR REDUCED OCCLUSION condition first and the VISAR MINIMAL OCCLUSION condition second. The keyboard outline was held constant across all conditions, as was the position and visual appearance of the *BACKSPACE*, *SPACE* and *DONE* keys.

Table 6.7 Entry rate (wpm) and character error rate (CER%) descriptive statistics from Experiment 3. Results show mean ± 1 standard deviation [min, max] for entry rate and median [min, max] for character error rate.

Condition	*Entry Rate (wpm)*	*Error Rate (CER)*
VISAR REDUCED OCCLUSION	10.39 ± 2.48 [7.02, 14.90]	2.04 [0.00, 4.57]
VISAR MINIMAL OCCLUSION	10.56 ± 2.59 [6.52, 15.26]	3.20 [0.28, 5.54]

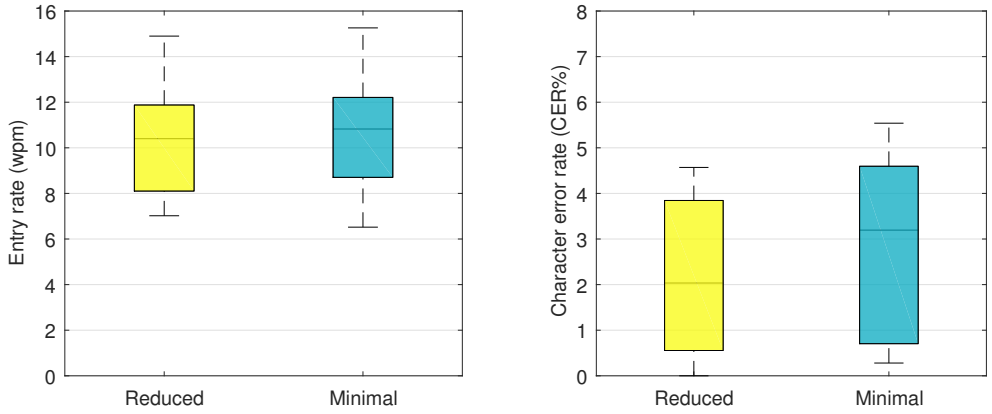


Fig. 6.17 Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 3. Entry and error rates are marginally higher in the minimal condition.

Users continued to receive visual feedback of the detected touch intersection with the keyboard plane via the small circular marker indicating the most recent touch location. In the VISAR REDUCED OCCLUSION condition, the nearest key outline would flash to green then fade back to white. No letter labels were shown in either condition in response to the basic touch event. Activation of the optional precision fall-back method would show the letter label and outline of the key currently in focus only. The key would fade back to its original visual configuration (depending on the condition) upon change of focus or deactivation of the precision fall-back mode.

6.8.2 Results

The key performance metrics for the two conditions in Experiment 3 are presented in Table 6.7. The difference in text entry rate between the reduced and minimal conditions is marginal and not significant ($F_{1,11} = 0.428$, $\eta_p^2 = 0.037$, $p = 0.526$). The reader is reminded that condition order was not balanced in this experiment and so the asymmetric learning effect represents a second plausible explanatory variable. The significance tests presented in this section thus reflect the coupled influence of minimising visual features and additional practice. The null

Table 6.8 Median questionnaire response in Experiment 3 on a five point Likert scale from 1-strongly disagree to 5-strongly agree.

Statement		*Median*
Q5	It was still possible to type effectively without key labels.	5.0
Q6	It was still possible to type effectively without key labels and outlines.	5.0

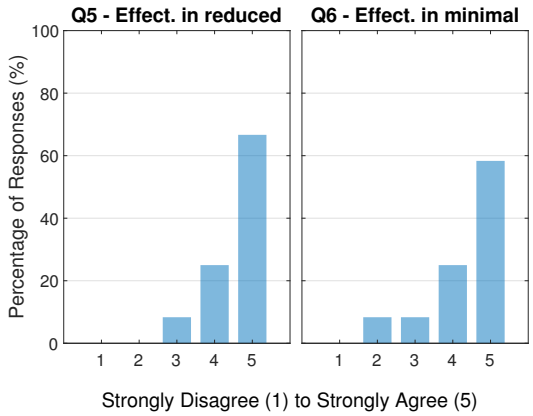


Fig. 6.18 Distribution of responses to Experiment 3 questionnaire statements Q5-6 as defined in Table 6.8.

result is still interesting as the practical application of a minimal occlusion interface would likely be introduced with a similar process of staged reduction of visual features. The entry rate result suggests that minimal visual features combined with additional practice yields performance largely indistinct from the keyboard with just key labels removed.

Interestingly, out of the 12 participants who completed the four conditions in Experiments 2 and 3, 10 achieved their highest entry rate performance in either the minimal or reduced occlusion configuration despite the lack of visual features. The maximum entry rate across all the conditions that comprised Experiments 2 and 3 was achieved in the VISAR MINIMAL OCCLUSION condition: 15.26 wpm with a character error rate of 0.35%. Figure 6.17 provides a boxplot of these performance metrics.

There was a small error rate difference between reduced and minimal occlusion, though not significant ($\chi^2(1) = 3.000$, $p = 0.083$). As a point of comparison, VISAR with no reduction in visual features in Experiment 2 yielded a median character error rate of 0.88%. Minimising field-of-view occlusion by removing visual keyboard features therefore does increase error rate, however, the resulting error rate with minimal occlusion is still below a tolerable threshold for CER (<5%).

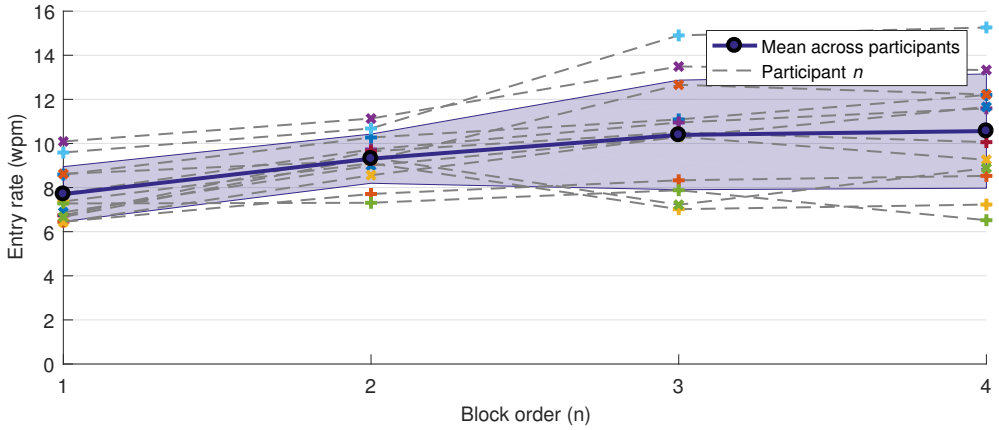


Fig. 6.19 Mean entry rate (wpm) across participants plotted in chronological order of test block in Experiments 2 and 3, irrespective of condition. Shaded region shows $\pm$ 1 standard deviation.

After completing Experiment 3, participants were asked to respond to a short questionnaire. The two statements Q5 and Q6 in Table 6.8 examined perceived typing effectiveness under the reduced and minimal occlusion conditions. The median responses are presented in Table 6.8 while the full distributions are shown in Figure 6.18.

Interestingly, participants were overwhelmingly positive in their self-assessment, with 91.67% either agreeing or strongly agreeing that they could type effectively without key labels. This proportion was only slightly lower for the statement corresponding to no key labels or outlines (83.33%).

The result that group mean entry rate was highest in the two conditions with the least visual features raises the question: to what extent does learning influence performance improvement? Figure 6.19 plots the mean entry rate for each condition performed during the single session that comprised Experiments 2 and 3. Recall that Experiment 2 balanced the order of presentation of the with and without fall-back conditions. This has been taken into account, such that Figure 6.19 shows tests results of each participant as they were performed in chronological order and irrespective of keyboard condition. The positive gradient observable across blocks 1 to 3 indicates the presence of a distinct learning effect and suggests a correlation between practice and performance.

6.9 Experiment 4: Design Iteration and Extended Use

This section describes several refinements of both the VISAR and BASELINE keyboard interaction methods and interface designs. The main design change involves the addition of word predictions based on the current input text. This design modification stems from *DP 4*

described in Section 6.4. The revised VISAR and BASELINE conditions are subsequently evaluated in an extended-use experiment in which participants are exposed to each keyboard condition over a 1.5 to 2 hour session. The revised conditions are subsequently referred to as BASELINE* and VISAR*.

6.9.1 Word Predictions and Decoder Refinement

The relatively low mean entry rates observed in the previous experiments (6 wpm in Experiment 1 and up to 10 wpm in Experiment 3) suggests that users are inherently rate limited by the two selection methods available. A potential strategy for improving entry rates is the provision of word predictions. Here, word predictions are defined as the presentation of the n most likely words based on the currently entered characters and sentence context. The user selects from among these presented word predictions to insert the word rather than typing all the remaining characters. Indeed, the system keyboard on the Microsoft HoloLens does provide word predictions and so the investigation of their potential effect is particularly relevant.

A trigram language model was integrated into both keyboard implementations. Preliminary testing indicated that a trigram language model provided comparable predictive power with the HoloLens system keyboard for the typical phrases used in the experiment. This trigram language model was trained in the same manner and on the same data as the 4-gram model used by the decoder. In the implementation described, three alternative word predictions were presented above the keyboard and updated as the user typed. If the user made a selection from among these alternatives, predictions for the next word in the sentence were shown.

The participant entry logs from Experiments 2 and 3 were used to further refine the decoder parameters. Furthermore, the decoder in VISAR* was extended to provide relevant word predictions even under input error. For example, a user might have typing errors in the literal interpretation of the current word's prefix. The algorithm provides the most probable word predictions taking into account the distribution over possible word prefixes and the language model probability of possible word predictions.

Typing a full word and entering a space on the VISAR* keyboard initiated the standard correction behaviour as described in Section 6.5.2: the most likely word given the observation sequence replaced the typed string. Importantly, however, when the decoder is activated by entering a space, the VISAR* keyboard temporarily re-purposes the three slots introduced to present the word predictions. The three next most likely decoder results are presented in these slots and can be selected to replace the inserted word. If the literal string typed is not already among these three alternatives, it is included and replaces the least likely of the presented words. This approach enabled users to select and re-insert the literal input in circumstances where the decoder incorrectly replaced the typed string.

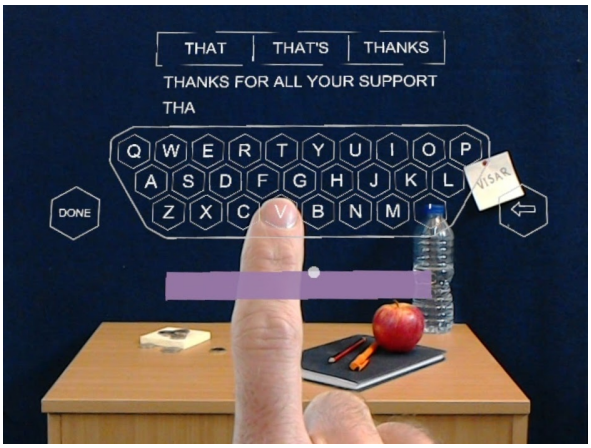


Fig. 6.20 The BASELINE* keyboard condition as viewed through the HoloLens.

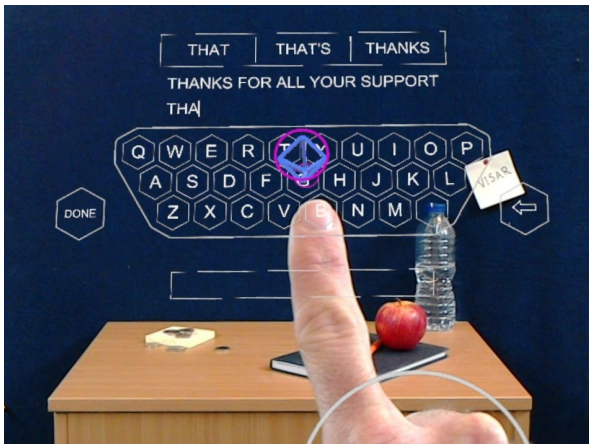


Fig. 6.21 The VISAR* keyboard condition as viewed through the HoloLens.

6.9.2 Interface Design Changes

The appearance of the BASELINE* and VISAR* keyboard conditions as seen through the HoloLens are shown in Figures 6.20 and 6.21 respectively. The following interface changes described were made based on observation of participants using the keyboard and informal qualitative feedback obtained during the previous experiments. Several participants observed that it was at times difficult to be close enough to reach the keyboard comfortably while maintaining a sufficient amount of the keyboard within view. Note that the Microsoft HoloLens provides a somewhat limited display window which means that near objects are prone to extending outside the render region and so appear cut-off. A decision was thus made to bring the keyboard closer to the user so that it was easier to reach, while also reducing it in size so that it was fully rendered. Adding the word prediction selection functionality also required that the position of the *BACKSPACE* and *DONE* keys be adjusted to make more efficient use of the available display region. These changes resulted in an apparent key diameter of approximately 17.5 mm and a key layout (keys *A* to *Z* plus apostrophe) of width 175 mm by height 52.5 mm.

During previous experiments, several participants also complained that typing on a vertically oriented keyboard plane was uncomfortable after extended use. It was suggested that the keyboard plane might be tilted and lowered to better map with how the hand traverses the space with minimal shoulder movement. This proved an effective suggestion and was incorporated into the revised design. The centre of the top keyboard row was thus positioned relative to the headset origin (a point approximately located slightly in front and above the user's eyebrows) with an offset of 500 mm away and 70 mm down and the whole layout was inclined at 20°.

The index cursor was also modified from a single flat circle to a pair of circles and a wireframe pyramid (compare the original cursor shown in Figure 6.6 with the revised cursor

shown in Figure 6.21). This change was made in response to feedback from participants that they had difficulty judging the depth of the original index cursor.

Note that the precision fall-back method introduced in Experiment 2 was also included in the VISAR* condition. It behaved in the same way as described in Section 6.7.1.

6.9.3 Method

In this experiment and in contrast with Experiment 1, the size and location of the keyboard was held constant in both conditions. This was done to reduce potential confounding effects associated with interactions between the selection method and the keyboard placement and/or sizing.

The test protocol was also revised from that used in the previous experiments to assess performance over blocks of a fixed number of phrases, rather than fixed time periods. This was done in part to encourage users to maintain high entry rates but also to ensure that participants would type the same number of phrases over the full session in both conditions. Participants would thus type 20 phrases per block for eight blocks to complete one session. Participants were encouraged to take a short break between each block. The eight blocks of 20 phrases that constituted a single session were all completed in the same keyboard condition. The two conditions were thus allocated to their own individual sessions, and were conducted on different days. Sessions were scheduled such that there was no more than one day break between each session. The order of conditions experienced by the participants was counterbalanced. Each session would typically last between 1.5 and 2 hours depending on participant typing entry rates. Participants were compensated with a £30 Amazon voucher for their time.

An introductory familiarisation task was also added to the experiment protocol to ensure participants achieved basic competence with the relevant selection method before beginning to type on the keyboard. The task serves to separate the device and interaction familiarisation from the keyboard familiarisation in order to better isolate the specific learning and performance effects associated with the two test conditions. The task involved selecting targets in a fixed sequence (a simplified circular target acquisition task). Participants were required to select all 10 targets within 15 s. If all targets were not selected within 15 s, the task would reset and repeat until this was achieved to ensure a minimum level of selection method proficiency was achieved. Following the familiarisation task, participants were then instructed to type five practice sentences. As in prior experiments, participants were encouraged during this practice period to ask questions and make sure they understood the interaction mechanism and keyboard functionality. Finally, one additional block was added to the VISAR* condition to investigate how predictions and extended use influenced performance in the minimal occlusion configuration.

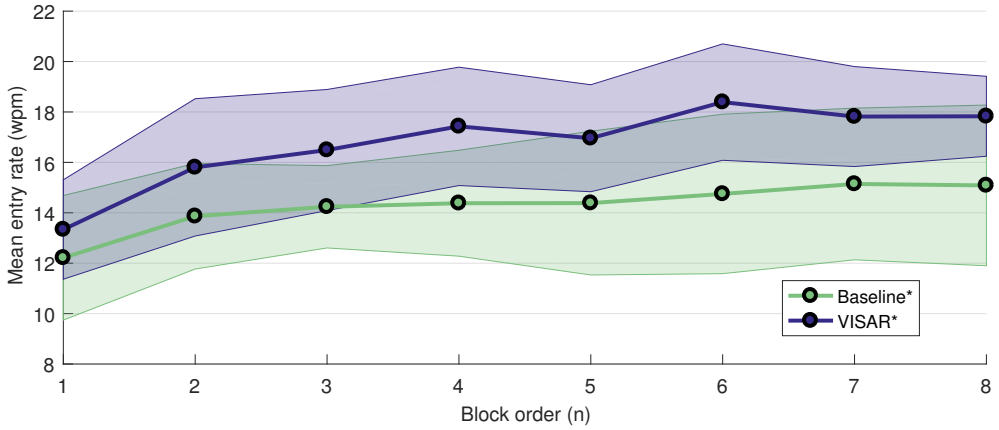


Fig. 6.22 Mean entry rate (wpm) across participants over the eight experimental blocks. Shaded region shows ± 1 standard deviation.

A total of 340 distinct phrases (20 phrases per block with eight blocks in the BASELINE* condition and 9 blocks in the VISAR* condition) were selected from the wider Enron dataset. These phrases were then randomly allocated over the conditions, blocks, and participants. Consistent with previous experiments, all phrases were constrained to be 40 characters or less, four words or more, and containing only the letters *A* to *Z* plus apostrophe. The out of vocabulary percentage for the phrase set in Experiments 4 was 1.18%. None of the phrases had been used in the previous experiments.

6.9.4 Results

A new group of 12 participants were recruited for the experiment (7 female, 5 male). None had participated in any of the previous experiments or had any experience with the Microsoft HoloLens.

Figure 6.22 shows the mean entry rate across participants over the eight blocks in each condition. The shaded region shows the standard deviation across participants in the given block number. The gradient is steepest in both conditions between blocks 1 and 2 then increases more gradually. This suggests a pronounced initial learning effect before a transition to a more gradual performance improvement with increased exposure and experience.

Table 6.9 shows the descriptive statistics for Experiment 4 following the completion of all eight test blocks. Entry rate and error rate results are also presented in Figure 6.23. The mean entry rate (wpm) over all eight blocks was 14.26 in the BASELINE* condition and 16.76 in VISAR*. Both methods achieved this with median character error rates (CER) under 1%. The difference in text entry rate over all eight blocks between the BASELINE* and VISAR* conditions is significant ($F_{1,11} = 9.014$, $\eta_p^2 = 0.450$, $p < 0.05$).

Table 6.9 Entry rate (wpm) and character error rate (CER%) descriptive statistics from Experiment 4. Results show mean ± 1 standard deviation [min, max] for entry rate and median [min, max] for character error rate.

Condition	*Entry Rate (wpm)*	*Error Rate (CER)*
BASELINE*	14.26 ± 2.12 [11.22, 18.24]	0.29 [0.02, 0.85]
VISAR*	16.76 ± 1.67 [14.43, 19.11]	0.51 [0.26, 1.72]

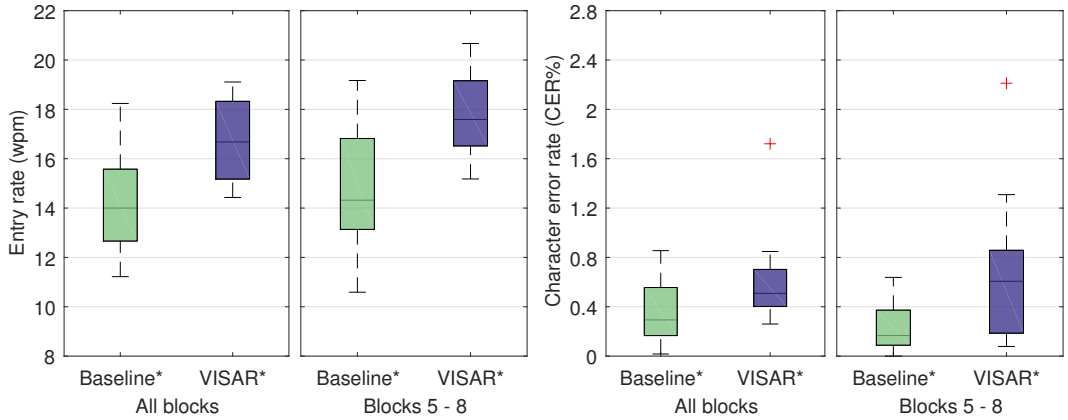


Fig. 6.23 Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 4. Plots show performance over all eight blocks as well as in just the final four blocks (blocks 5-8).

Another relevant point of analysis is the performance difference once the dominant learning effect has subsided. This is examined by computing entry rates in the final four blocks only. Figure 6.23 also presents boxplots based on the final four blocks only. The mean entry rate in the final four blocks was 14.84 and 17.75 for the BASELINE* and VISAR* conditions respectively. This represents a speed increase from the baseline of 19.6%. This difference is significant ($F_{1,11} = 8.237$, $\eta_p^2 = 0.428$, $p < 0.05$).

The highest mean entry rate for a given 20 phrase block for VISAR* was achieved by participant 7 during block 6 with 23.38 wpm at a character error rate of 0.24%. The best performing block for BASELINE* was participant 2, also in block 6, with 20.55 wpm at an error rate of 0.00%.

In addition to net entry rate, it is useful to examine the underlying efficiency of both selection methods. An interesting first point of comparison is the number of repetitions required to complete the initial target acquisition familiarisation task. Recall that this task required participants to select 10 targets appearing at opposing points on a circle of fixed radius within 15 s. The mean number of executions in the gaze-then-click and touch based selection

techniques were 7.3 and 2.2 respectively. These results are distorted by some participants who found the air-tap gesture particularly difficult to perform reliably. The median number of repetitions, 5 for gaze-then-click and 2 for touch, is perhaps a better reflection of the relative efficiency of the two techniques. This result is likely a consequence of two key factors: i) the touch driven interaction technique exploits a familiar paradigm allowing users to apply already established motor skills; and ii) discrete target selection is inherently more efficient and reliable without the use of a hand gesture. The first factor was observed to some degree in both Experiment 1 and Experiment 4, where a steeper learning effect was observed in the VISAR conditions. The second factor is further explored in more detail below through an analysis of input efficiency based on Fitts' law.

Typing on a keyboard may be abstracted to a sequential target acquisition task. Fortunately, it is possible to leverage established analytical approaches to estimating the underlying qualities of a selection technique that are scale independent.

The key log data collected during this experiment was post-processed to extract the key-to-key transitions. Only those input phrases that contained no interim selection errors were included in this analysis. The key-to-key transitions form the basis for estimating throughput according to Fitts' law.

Fitts' law predicts that Movement Time (MT) in making a selection is linearly proportional to Index of Difficult (ID), a non-dimensional metric representing the difficulty associated with a selection.

Thus, movement time can be defined as

$$MT = a + bID, \tag{6.2}$$

where a and b are regression coefficients and ID is (according to the Shannon formulation of Fitts' law)

$$ID = \log_2\left(\frac{D}{W} + 1\right), \tag{6.3}$$

where D is the movement distance, and W is the target (key) width.

The next step is to extract the movement times associated with the different key transitions encountered while typing (only considering keys A to Z and apostrophe). The regular tessellation of the keyboard layout means that there are only 23 distinct key-to-key movement distances. Furthermore, the key sequences corresponding to two of the extreme movement distances (e.g. Q to P and A to apostrophe/Z to P as well as their inverses) do not occur in the stimulus phrase set. This leaves 21 distinct key-to-key movement distances which form the basis for the ID sample points (W is constant).

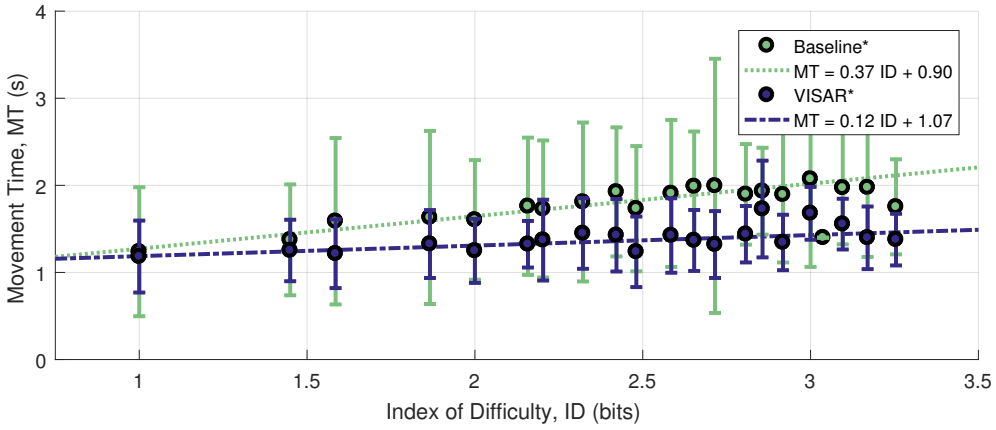


Fig. 6.24 Movement Time (*MT*) versus Index of Difficulty (*ID*) based on key transitions encountered while typing. Error bars show $\pm$ 1 standard deviation. The dashed lines show a linear regression of the two conditions.

Figure 6.24 shows the movement time versus index of difficulty for the two keyboard conditions. Throughput is computed as $TP = 1/b$, where b is the gradient of the regression line. The computed throughput was 8.26 bps for VISAR* and 2.68 bps for the BASELINE*. The time taken to make a discrete key selection is clearly an emergent property and influenced by multiple factors such as the human perception, motor control, and processing systems as well as device attributes such as tracking accuracy and latency. It is reasonable to assume that many of these factors are consistent across both conditions and this is corroborated by the similar intercept values, a, shown in Figure 6.24. The difference in slopes (and hence difference in throughput) visible in Figure 6.24 indicates that the negative effect of increasing task difficulty (*ID*) is lower in the VISAR* condition. In other words, participants could select distant keys nearly as well as nearby keys in the VISAR* condition whereas the BASELINE* condition saw a more prominent negative effect as the distance between keys increased. This may stem from inherently superior motor control of the hand in contrast to head movements. The throughput values reported here should, however, be interpreted with caution given that the free-form typing task does not closely replicate the traditional protocol used in a typical Fitts' law experiment. Furthermore, constraining the analysis to only phrases with completely accurate selections likely inflates the computed throughput values. Nevertheless, the relative magnitude of the two values does provide an indication of the comparative efficiency of the two selection methods.

The character error rates were significantly higher in the VISAR* condition ($\chi^2(1) = 5.333$, $p < 0.05$), although the mean character error rate over all eight blocks was less than 1% for 11 of the 12 participants. Error rates of this magnitude are typically considered tolerable in

most text entry tasks. Nevertheless, this result does highlight the speed-accuracy trade-off typically observed in alternative text entry methods. The specific usage scenario may dictate whether a user is willing to accept a higher error rate for the sake of higher entry rates. It also highlights the importance of error correction and error prevention functionality being considered in parallel with underlying keyboard and interaction design. For example, it was observed that entry errors would occasionally occur when users failed to check the word returned after a decode. Improving the visibility of the decode process through careful interface cues may thus help to further reduce these types of errors.

Although the precision fall-back method was available in the VISAR* condition, it was only used by five of the 12 participants. Among these five participants, the fall-back method was used on average 3.0 (median = 1.0) times on distinct words. This rate of usage is a distinct reduction from that observed in Experiment 2, where the average usage per participant was 5.1 distinct words and a median of 2.5. Only one usage of the fall-back method was a response to a decoding failure that the user sought to correct. All other intentional usages of the fall-back method were pre-emptive in that participants had not experienced a prior decoding failure on that word. The reduction in usage of the fall-back method is likely a consequence of both the introduction of word predictions and a reduced explicit emphasis on the feature within the experimental briefing and protocol. Furthermore, it was observed that in instances where a possible decode failure was anticipated, several participants found that simply taking more time and care to hit the desired keys was sufficient to correctly type the word.

Participants completed a post-experiment questionnaire targeting impressions of their typing speed, accuracy and comfort under the two keyboard conditions. The questionnaire statements responded to, and their median responses are presented in Table 6.10. Note that participants were asked to exclude their experience of the minimal occlusion condition described in the following section when considering their assessment. Figure 6.25 presents the full distribution of responses to the statements in Table 6.10. No distinct difference is apparent in terms of typing speed ($Z = -1.730$, $p = 0.084$) or accuracy ($Z = 0.741$, $p = 0.458$). Participants were generally less willing to agree with the statement that VISAR* was comfortable although the difference from the BASELINE* was not significant ($Z = 1.801$, $p = 0.072$). From inspection of Q3 in Figure 6.25, a bimodal distribution of responses for VISAR* is observable. This result is consistent with informal comments from several participants that the VISAR* condition caused some discomfort for the shoulder. The final question on the questionnaire asked participants to indicate a preference between the two conditions. VISAR* was preferred by eight of the 12 participants (67%).

As described in Section 6.9.3, participants were exposed to one additional block in the VISAR* condition where the visual features of the keyboard were set to the minimal occlusion

Table 6.10 Median questionnaire response in Experiment 4 on a five point Likert scale from 1-strongly disagree to 5-strongly agree.

Statement		BASELINE*	VISAR*
Q1	The keyboard made it easy to type quickly.	4.0	4.0
Q2	The keyboard made it easy to type accurately.	4.0	3.0
Q3	The keyboard was comfortable to use.	4.0	2.5

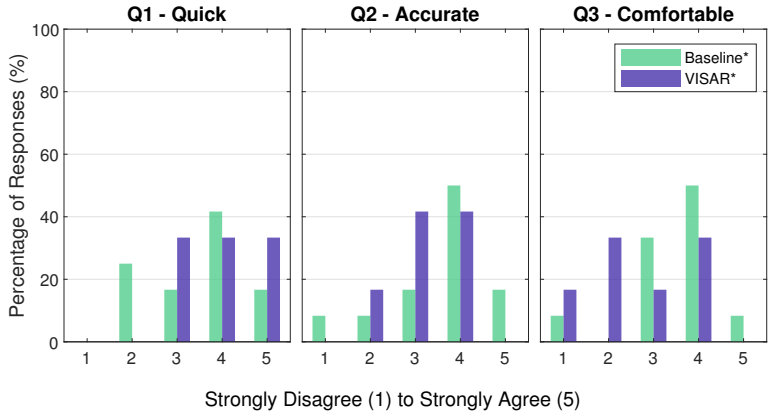


Fig. 6.25 Distribution of responses to Experiment 4 questionnaire. The question statements Q1-3 are defined in Table 6.10.

configuration, i.e. no key outlines or key labels were shown. Figure 6.26 plots entry rate versus error rate for all 12 participants under this configuration. The performance of each participant in the immediately preceding eighth block of the full visibility VISAR* keyboard condition is also shown for comparison.

The results presented in Figure 6.26 highlight a distinct split between participants who were unable to effectively use VISAR* in the minimal occlusion configuration and those who were largely unaffected by the removal of key outlines and labels. Seven of 12 participants experienced a reduction in entry rate of between 10.0% and 61.9% against their previous block performance combined with a signification deterioration in character error rate. The other five participants maintained a mean entry rate between 4.4% slower and 17.9% faster than their previous full visibility block while all had character error rates of less than 1.2%. It is suspected that this observation is likely to be correlated with a participant's ability to effectively touch type. It is worth noting, however, that this same distinct split was not observed in Experiment 3. Unfortunately, the pre-experiment experiential survey did not capture touch typing ability and so investigation of this correlation remains as future work.

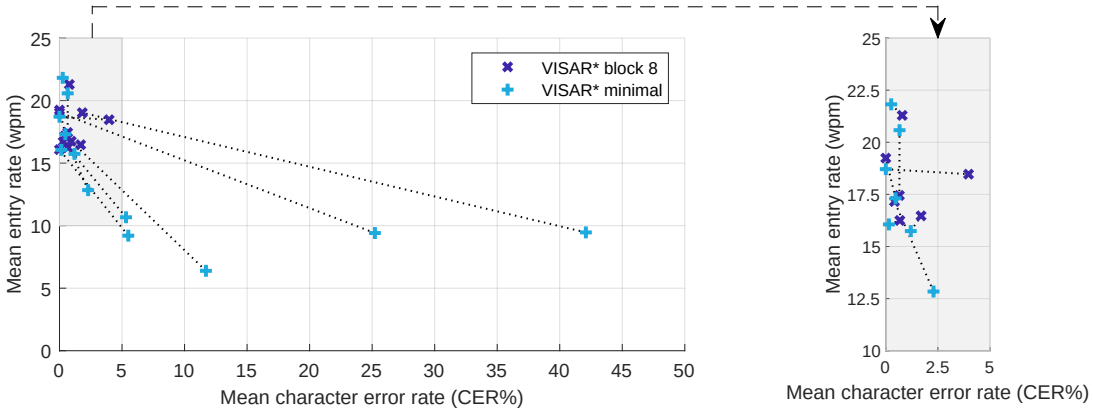


Fig. 6.26 Mean entry rate (wpm) versus mean character error rate (CER%) for VISAR* in the minimal occlusion configuration and the final block (block 8) of the standard visual configuration. Dashed lines link the two results of individual participants. The left plot shows data points for all participants. The right plot provides an enlarged detail view of the shaded region shown on the left. Only data points for participants with error rates under 5% are shown in the enlarged detail view. Five participants shown in the enlarged detail achieved entry rates comparable with their prior performance, despite the lack of key outlines and labels.

The introduction of word predictions is speculated to have altered the typing strategy of some users which inadvertently primed them differently for the minimal occlusion block in Experiment 4. The provision of word predictions allows users to obtain near instantaneous feedback on the decoder's best estimates of their intended input. When users can see the keys, they are more likely to touch on or near the intended key. This improves the likelihood that the presented predictions will include the intended word, even if there have been only two or three touches. Under higher levels of input noise such as encountered in the minimal occlusion configuration, more data points (touches) may be required to accurately predict the intended word. However, users accustomed to seeing their intended word among the predictions after very few touches have not built up sufficient trust and confidence in the decoder to continue typing despite apparently erroneous predictions. In contrast, participants in Experiments 2 and 3 were not shown predictions and so were more likely to type out the full word and rely on the decode step to correct errors. Positive examples of successful error correction served to reinforce trust and confidence in the decoder. This theorised interaction between interface features and typing strategies requires further investigation. Nevertheless, it was observed that VISAR* did enable a subset of the participant group to maintain their entry rate under the minimal occlusion configuration.

6.10 Validation Study: Spatialised Text Entry

This validation study examines whether VISAR* is a suitable text entry method for typical AR applications. A short experiment, chiefly examining user experience and behaviour, was designed to expose participants to a range of short text entry tasks under conditions relevant to head-mounted AR.

6.10.1 Method

Four participants were recruited for the study (4 male). The experiment session lasted for approximately one hour and participants were compensated with a £10 Amazon voucher.

Participants received an introductory briefing before performing the same target acquisition familiarisation task described in Section 6.9.3. Participants would then also complete five practice phrases while seated (in the same arrangement as described in Section 6.9.3).

The main exercise in the experiment then required that participants explore the space where they would encounter four different kinds of text entry sub-tasks. Five of each sub-task were presented at locations dispersed throughout the space, resulting in 20 sub-tasks in total.

The four sub-tasks are described below:

- **TRANSCRIPTION**: A short phrase (taken from the Enron dataset) was printed on a page and attached to the wall at the task location. Participants were instructed to transcribe the phrase exactly.
- **DESCRIPTION**: A simple illustration was printed on a page and attached to the wall at the task location. Participants were instructed to describe the image, e.g. one image showed an image of a man and a woman riding a tandem bicycle.
- **MESSAGE**: A 'message' would be received at specific locations in the space asking a simple question. Participants were instructed to respond to the question, e.g. one message asked, 'What did you have for breakfast?'
- **ANNOTATION**: Participants were instructed to pick an object at the task location and annotate or describe it, e.g. one free annotation task was located where several portable fire extinguishers were located.

Upon finding a sub-task in the space, participants would select the corresponding virtual marker as shown in Figure 6.27. This would then bring up the keyboard (see Figure 6.28) allowing them to complete the task. Participants were encouraged to use at least four words

when crafting their text in the three sub-tasks involving composition. Upon selecting *DONE*, the entered text would be submitted and the keyboard would close.

Participant entry rates were recorded. Following the experiment, participants were also requested to complete a System Usability Scale survey [16] targeting their experience of the system as a whole. A short semi-structured interview was also conducted to obtain participant feedback on the user experience in the spatial annotation exercise.

6.10.2 Results

The results of this experiment should be interpreted with some caution given the limited number of participants involved. The quantitative results are presented to provide an indication of what might be achievable rather than as an attempt to describe typical user performance.

Figure 6.29 presents boxplots of the entry rates achieved by the participants in the four different sub-tasks as well as over all sub-tasks irrespective of task type. As might be expected, the different sub-tasks resulted in different entry rates. Intuitively, the highest mean entry rate achieved was in the TRANSCRIPTION task (10.32 wpm). The overall mean entry rate achieved is 8.74 wpm. This is clearly considerably lower than the maximum rates archived in Experiment 4 but tolerable for a casual text entry method. Furthermore, it is known from the results of Experiment 4 that entry rates improve significantly with practice. The participants in this experiment only completed five practice phrases before commencing the spatial annotation task.

The four participants scored the system using the System Usability Scale (SUS): 72.5, 75.0, 77.5 and 67.5 (mean 73.1). Bangor et al. [8] suggest that an SUS score above 70 indicates acceptable system usability. The SUS ratings provided by the participants are thus very

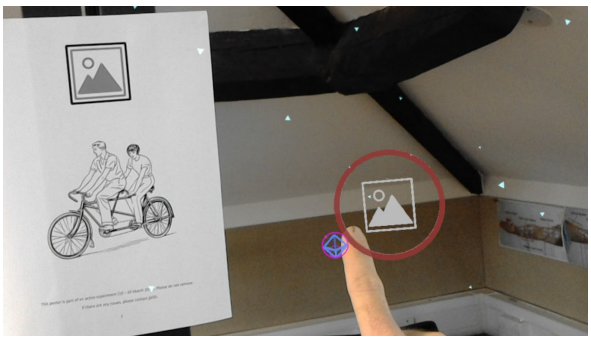


Fig. 6.27 A DESCRIPTION sub-task requiring the user to describe an image in four words or more.

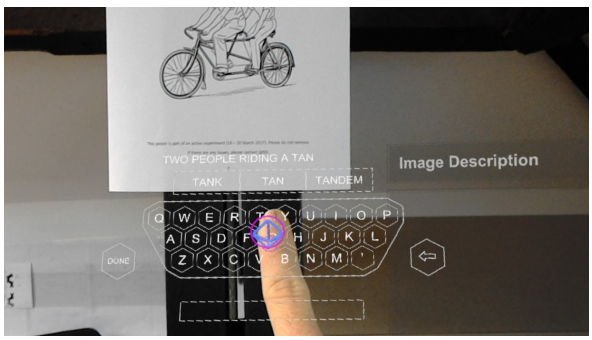


Fig. 6.28 The keyboard appears after selecting the task, allowing the user to enter the description.

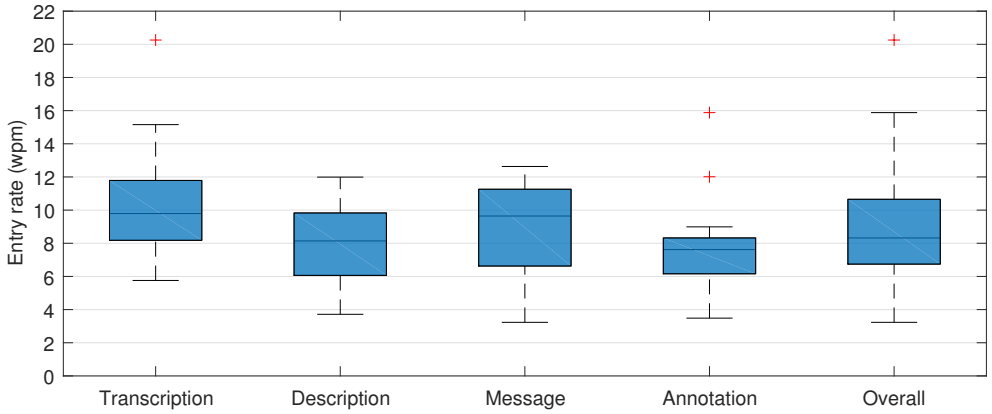


Fig. 6.29 Boxplots of entry rate (wpm) for each sub-task and over all sub-tasks in the validation study. Red crosses indicate outliers based on $Q_{1/3} \pm 1.5 \times (Q_3 - Q_1)$.

promising and appear to indicate the viability of the VISAR keyboard as a text entry method for AR HMDs.

Following the experiment, participants were asked to comment on the aspects of the task that they enjoyed. Responses focused on the positive experience of being able to freely and interactively explore the space ('I liked the fact that it was a mixed reality environment so you had some elements that were virtual and some that were real', 'It was good to explore things') as well as the helpfulness of the directional cues provided ('It was good that it was directing you towards where it was going to ask something and then you are communicating through the system', 'The hint it gives me to find the tasks and also the sound system, it reminds me that there is a message coming in. That's the part that I really liked').

Participants were also asked to comment on any aspects of using the system that they found annoying. Two of the participants commented on comfort, one referring to the shoulder discomfort ('The main thing was the comfort of holding your hand up and especially the finger up as well') and the other referring to general discomfort with the headset ('I think the first problem is that it is not comfortable to wear the headset'). Other issues identified as being annoying were the small size of the headset display region and the lag in hand tracking. Also related to hand tracking, participants were warned in the introductory briefing that the HoloLens reported hand location was less robust when used in close proximity (<0.5 m) to walls or other fixed physical objects. Some of the participants encountered this issue during the task and commented on this in the interview. The keyboard location could, however, be adjusted by looking away from the current location and re-focusing on the new desired location. Participants were thus able to quickly remedy this issue when encountered.

In other miscellaneous comments, one participant observed that, 'I did find that you get drawn into the virtual elements and I had to stop and realise what was around me sometimes.' This observation highlights a broader challenge for AR interface design in terms of managing cognitive tunnelling. Another observation related to the lack of a physical response when interacting with the virtualised input surface of the keyboard, 'I feel like there is auditory and visual feedback but there is no tactile feedback on your finger and it is a bit disconcerting.' Examining the potential for minimal tactile feedback to enhance the experience of interacting with virtual objects in AR, and especially for text entry, offers an exciting avenue of exploration.

Finally, the four participants were asked whether the system worked sufficiently well to effectively complete the task. All answered in the affirmative, thus complementing the SUS score results obtained.

6.11 Discussion

The five experiments presented in this study represent a structured attempt to apply immersive, and more specifically, AR-focused design principles to the challenge of supporting efficient text entry for AR HMDs. Experiment 1 sought to test the hypothesis that the more natural direct-touch technique would support higher entry rates than a gaze-then-gesture baseline. The results revealed only a marginal improvement in overall entry rate associated with the VISAR keyboard although the time taken to select discrete keys was significantly faster. This finding encouraged the subsequent exploration of why the more rapid key selection using the VISAR system failed to directly translate into higher entry rates.

An obvious deficiency identified was the lack of a precision fall-back mechanism allowing users to specify that certain letters are inputted with full certainty and need not be changed by the decoder. The lack of this feature resulted in significant time being wasted on correcting incorrect decoder returns. Therefore, a seamless high-precision fall-back mechanism was designed and evaluated in Experiment 2. This experiment indicated that the provision of a fall-back method in VISAR does not adversely affect entry rate and can help to reduce the error rate when employed effectively.

Experiment 3 is motivated by the design objective in AR HMDs to minimise keyboard occlusion of the real-world (*DP 3* in Section 6.4). Two conditions were examined which sequentially removed visual features from the keyboard layout: first no letter labels, then no letter labels or key outlines. Almost all users in the experiment expressed surprise at being able to type effectively without key labels or outlines. The fastest participant in Experiment 2 and 3 achieved an entry rate of 15.26 wpm at an error rate of 0.35% in the no keys or outlines condition shown in Figure 6.16. In general, it was found that users could type quickly using

no visual features and although error rates rose significantly, the absolute error rates with no visual features were still far below the maximum tolerable threshold for character error rates (typically set at 5% CER). This finding demonstrates how VISAR can provide superior text entry support for AR HMDs.

The design of VISAR was further iterated upon to improve comfort in use and to include probabilistic, error-tolerant word predictions as suggested by *DP 4* in Section 6.4. Experiment 4 evaluated the refined VISAR keyboard against a similarly improved baseline condition, again based on the gaze-then-gesture selection paradigm. The results show that VISAR was capable of producing a mean entry rate across participants of 16.76 wpm compared with 14.26 wpm when completely naïve users were exposed to each method and requested to type for between 1.5 and 2 hours. With the dominant period of learning removed, the mean entry rates were 17.75 wpm and 14.84 wpm for VISAR and the baseline respectively. This finding suggests a significant speed advantage for VISAR of approximately 20% relative to the baseline. The highest mean entry rate among all participants over a distinct experimental block of 20 phrases was 23.38 wpm at a character error rate of 0.24%. More generally, the character error rate of the VISAR keyboard was elevated against the baseline condition with a mean of 0.63% across participants. While this is within generally tolerated levels, the result does highlight a speed-accuracy trade-off. The additional speed provided by the VISAR keyboard comes with the cost of higher error rates.

Finally, the validation study demonstrated the suitability of VISAR for typical mobile AR text entry tasks. Participants were able to achieve mean entry rates in the range of 7 to 10 wpm in a variety of transcription and composition tasks that were encountered while walking to explore a physical space. These entry rates were achieved after only a brief introduction on the use of the system and a seated training period involving only five practice phrases.

6.11.1 Implications for Design

Although this study demonstrates the effectiveness of the VISAR technique, no claim is made that the entry rates, even at the maximum achieved by a participant in a distinct test block (23.38 wpm), should be considered ‘fast’. Nevertheless, the obtained entry rates are sufficient for casual text entry in an AR HMD environment when a physical keyboard or other human-machine interface device is unavailable. The typical error rates are well within tolerable levels for most casual text entry tasks. The main contribution of this chapter is in highlighting several unique design requirements and design principles relevant to AR HMDs.

The direct-touch interaction method exploited in VISAR is based on relatively coarse hand tracking. Despite this, users quickly adapted to the task of controlling the index cursor to touch keys, despite its positioning lag and inability to reflect hand articulation. Although the

influence of high-precision hand-tracking in this task is worth exploring, the fact that sub-optimal tracking with a state-of-the-art AR HMD can still be exploited to deliver an immersive interaction experience for text entry should not be overlooked.

This study has also highlighted several design considerations specifically relevant to text entry in AR. First, participants found the interaction method tiring on the arm. This is not surprising given the duration of several of the experiments. In practice, however, it is envisaged that the likely use cases of fully hands-free environment-embedded AR text entry are sufficiently sporadic and short to make the VISAR approach worthwhile.

A second observation is that users appeared more prone to mishitting keys lower in the keyboard due to the trajectory followed by the hand during reaching. This has potential implications on the design of the keyboard layout and potential placement of other interaction elements.

The unfortunate inverse relationship between keyboard size and presentation proximity enforced by the constrained display window in current AR HMDs also introduces difficulties in accommodating a wide range of users. Although this statement requires proper investigation to be conclusive, it was casually observed that participants with shorter arms struggled more to find a comfortable position in which they could see a sufficient amount of the keyboard and easily reach the keys.

In terms of error-tolerant touch-based decoding, additional design explorations can possibly improve performance in an AR HMD environment, where interactions are laborious and/or imprecise. Deficiencies in the word-by-word decoding approach were observed in instances where there are words closely located in the feature space, e.g. so and do, out and put, toy and you. This is particularly problematic in cases where such words occur early in sentences where the left context does not help narrow the search. A further challenge in deploying advanced decoding is about educating users. Many users expressed surprise at the effectiveness of the decoder but had to be encouraged to leverage its capability in order to increase their entry rate.

6.11.2 Limitations and Future Work

While the study reported here seeks to provide a thorough overview of the initial design and evaluation of a text entry method specifically designed for AR HMDs, several limitations and opportunities for future work are acknowledged.

Experiments 1 and 4 both highlight a speed-accuracy trade-off for the VISAR keyboard with the higher entry rate also producing higher error rates. Further work is required to determine whether suitable error correction and error prevention functionality might help mitigate this without degrading entry rates.

To test the applicability of the text entry method to fully featured text entry, it will be necessary to evaluate the effect of providing the full complement of punctuation as well as case modification. Furthermore, inclusion of insertion and editing functionality into the keyboard and interaction method would be necessary for a commercial product and also opens up many avenues for future work on investigating efficient correction interfaces for text entry in AR HMDs.

The system may lead to some discomfort during prolonged use due to the need to point with the hand without receiving any force feedback. To mitigate this, it is important to study the effect of keyboard pose and size on user comfort, possibly by employing metrics such as consumed endurance [65]. This finding also raises the question as to what text entry uses cases are likely to emerge for AR HMDs and what factors might encourage the user to transition between different input modes.

Related to this prior point is the informal observation noted in the previous section that users with shorter arm lengths may have experienced more difficulty with the direct touch interaction than those with longer arms. This observation suggests a potential effect of physiology on typing behaviour and performance. A future experimental investigation should therefore examine the influence of physiology on mid-air text entry, and specifically the influence of arm length. This investigation may be paired with an exploration of strategies for accommodating physiological differences among users.

The result that users can type effectively without any key outlines or labels suggests the system might also support text entry while walking. However, this would need to be carefully evaluated, updating findings from similar investigations for mobile phones [137].

Finally, the system architecture is flexible enough to support decoding of alternative text entry modalities, in particular gesture keyboard decoding [89]. Future work could investigate if this would result in any additional performance gain.

6.12 Conclusions

Many anticipated applications of AR require the ability to enter text. Text entry methods for AR should exploit the unique advantages of immersive interfaces rather than being cobbled together from paradigms borrowed from two-dimensional interfaces. This chapter examines the design of an augmented reality text entry method based on error-tolerant mid-air touch interaction with a virtual keyboard. Its effectiveness on the Microsoft HoloLens is investigated in a series of five controlled user studies.

The experimental results show that users can select keys more quickly using the direct-touch approach than with the gaze-then-gesture approach. This delivers significantly faster entry

rates when combined with probabilistic word predictions. A particularly striking result is that a sub-group of users can maintain and exceed entry rates when all key labels and outlines are removed from the keyboard so that only the keyboard region outline remains.

The key contributions of this chapter are three-part. First, six design principles informed by the literature and prior interface design experience are presented. These inform the design of productive text entry methods for AR HMDs. Second, a novel keyboard system specifically adapted to AR HMDs based on an error-tolerant touch-driven interaction paradigm and incorporating an inferred-to-literal fall-back method is demonstrated. It also supports configurable occlusion settings to improve user visibility of the physical environment. Third, empirical results from a comparison with a gaze-then-gesture baseline entry method, and an investigation of the influence of various design decisions are presented. This establishes a useful point of reference for future studies seeking to explore productive text entry in AR. In summary, this chapter shows that VISAR can support productive text entry on a head-mounted augmented reality display. It is hoped that the design principles upon which the system is based inspire other novel and efficient entry methods for AR.

6.13 Research Question 3 and the Design Process

This chapter has highlighted the value of inference in accommodating the high levels of input noise typically encountered in MR applications. The VISAR system demonstrated acts as a vehicle for answering *Research Question 3*, which is the focus of this chapter: *How can probabilistic inference be exploited to accommodate high levels of input noise in mixed reality applications to deliver more efficient interactions?* The concise answer to this question is that certain HCI tasks exhibit behaviours well suited to predictive modelling, and in such cases, well-framed inference efforts can readily disambiguate noisy user input. The VISAR system demonstrates how this approach paired with careful design of the interaction strategy can deliver improved performance in a mixed reality setting.

This chapter serves as a near complete illustration of the emerging design process described in Section 2.3. The four stages in this process were undertaken to both deliver an effective text entry system as well as to obtain understanding of the key determinants of performance from a design perspective. Significantly, this chapter extends the efforts presented in Chapter 4 by both examining the sensitivity of the various identified design principles and ultimately validating the complete system in Section 6.10.

Chapter 7

Probabilistic Optimisation

As the previous chapters have illustrated, designing interfaces for novel mixed reality applications is challenging. Typically, designers rely on experience or subjective judgement in the absence of analytical or objective means for selecting interface parameters. While such an approach may be sufficient in many cases, there are circumstances in which more rigour and structure is necessary. This chapter explores *Research Question 4: How can the unfamiliar and high dimensional design space for mixed reality applications be efficiently explored and refined through probabilistic optimisation?* The perspective brought to this research question differs from the previous three in that the focus is on the design process rather than the application itself.

This chapter demonstrates Bayesian optimisation as an efficient tool for objective interface feature refinement. This is a probabilistic optimisation approach that leverages an iteratively refined model of user performance for a given interface design parameterisation. This case study shows how Bayesian optimisation can be paired with crowdsourcing to rapidly and effectively assist interface design across diverse deployment environments.

The first experiment in Section 7.6 evaluates the approach on a familiar 2D interface design problem: a map search and review use case. Adding a degree of complexity, the second experiment in Section 7.7 extends the first by switching the deployment environment to mobile-based virtual reality. The approach is then demonstrated as a case study in Section 7.8 for a fundamentally new and unfamiliar interaction design problem: web-based augmented reality. Finally, Section 7.9.1 shows how the model generated as an outcome of the refinement process can be used for user simulation and queried to deliver various design insights.

7.1 Introduction

Without general guidance or analytical frameworks, user evaluation is critical to informing interface design. Performing this evaluation efficiently and identifying an optimum configuration is a fundamental goal of HCI. However, the process of optimising the user interface is a non-trivial exercise given the typically noisy behaviour of users and variability between users. Allowing the user to play a role in the optimisation process is an attractive solution, particularly in instances where each evaluation is inherently user driven and the application is susceptible to user variability.

This chapter examines Bayesian optimisation as a potential tool in performing interface optimisation in large scale, noisy user environments. Specifically, Bayesian optimisation is evaluated as an approach for online refinement of interface features through crowdsourced user participation. Bayesian optimisation is applied to refine the parameters that determine the visual features and interaction behaviour of a typical interface.

Two illustrative experiments to demonstrate the process and its flexibility are presented: 1) design of a 2D map search and review interface (such as encountered on a hotel booking site); and 2) design of a novel VR based search interface for the same task. These tasks and interfaces serve as a simple and familiar example to demonstrate the approach. Both interfaces are parameterised according to five design dimensions. Users are recruited through crowdsourcing to identify ideal values for these parameters. The Bayesian optimisation approach informs the selection of new test parameter values based on prior user performance, measured as the time to complete a discrete map search task. The experiment is structured to incorporate prior user data in batches in order to more clearly demonstrate an improvement in interface performance over time. As a baseline comparison, the Bayesian optimisation approach is compared with designs uniformly sampled from the bounded design space. This approximates arbitrary parameter settings chosen by a naïve designer.

In addition to the two experiments described, the procedure is applied to an even more challenging mobile based augmented reality case study. This provides a practical demonstration of the approach and highlights its flexibility.

The key contributions of this chapter are:

1. An evaluation of Bayesian optimisation for interface design refinement in two challenging design spaces.
2. A demonstration of the approach in a highly novel web-based AR design case study.
3. Implementation guidance for crowdsourcing interface design refinement using Bayesian optimisation.

7.2 Related Work

The broader challenge of actively supporting interface design refinement has been approached from a variety of perspectives. These efforts largely fall into three categories: model-based optimisation, post hoc refinement, and online refinement. Model-based optimisation methods support the designer at design time based on predictive models of the user [50, 159, 177, 142]. Keyboard layout optimisation is a popular application of this approach. Applications such as MenuOptimizer [7], DesignScape [133] and Sketchplore [177] demonstrate how these approaches can also be explicitly embedded into design support tools.

Post hoc refinement is an offline strategy in which collected data is either used directly or fed to user models to refine the interface. Clearly this encompasses the much broader workflow of making design changes based on feedback and controlled experiments [84]. More relevant to the context of this chapter, however, are efforts that formalize this approach [178, 105, 157]. Salem [157] demonstrates a structured approach to comparing and refining web landing page design alternatives using genetic programming while Liu et al. [105] explore optimal representations for mathematics pedagogy.

Online interface refinement, the category in which this investigation falls, describes methods which actively change the interface based on some objective during or between interactions. This approach is readily applied in games where an optimal performance or engagement level might be achieved through game feature refinement [111, 108, 77]. Similarly, BIGnav [103] probabilistically fused inputs and prior information about locations on a map to improve navigation performance. Online refinement has also been explored in psychology (e.g. [126]) to obtain maximally informative experiments. In the context of interaction in virtual environments, Octavia et al. [131] describe a conceptual framework for adapting interactions to user preferences.

Bayesian optimisation has significant potential in supporting this third strategy of interface refinement. Bayesian optimisation is a machine learning technique that facilitates the exploration of cost functions that are complex or can only be estimated by making noisy observations of a latent function. The approach is particularly useful when evaluation of the cost function is expensive: e.g. slow computational models, or evaluations that involve a physical process. A detailed review of the approach and practical applications of Bayesian optimisation is provided by Shahriari et al. [165]. Bayesian optimisation has been applied in user interfaces for a range of applications. Brochu et al. [15] applied the technique within a preference gallery to allow users to evaluate alternate settings for rendering smoke. Other applications have involved maximising user engagement in games [77], and optimising individual user settings for a hearing device [129]. Snoek's doctoral thesis [169] provides a comprehensive investigation of Bayesian optimisation for assistive technologies.

The incorporation of a Bayesian optimisation approach into interface design by exploiting user interaction data is challenging due to typically high noise levels. Running large numbers of users through an interface that may be poorly designed in its first iteration can also be difficult for many designers. Fortunately, crowdsourcing has emerged as a viable source of large volumes of users willing to undertake interface testing in return for compensation. Crowdsourcing can offer large quantities of data at low cost [82]. Comparative studies, such as those by Heer and Bostock [63], have demonstrated crowdsourcing as a fast and effective method for gathering graphical perception data providing results consistent with in-lab studies. There is also good precedence in interface research carried out using crowdsourcing [178]. Further, work by Komarov et al. [86] has shown that performance evaluation of user interfaces, carried out using both crowdworkers and in-lab participants, yields equivalent relative differences between experimental conditions.

Crowdsourcing has been employed at the intersection of interface design and Bayesian optimisation to efficiently collect large numbers of user interactions. Koyama et al. [87] demonstrate the potential of Bayesian optimisation to assist with visual feature optimisation. They decompose the higher-order optimisation problem into one-dimensional line searches that can then be allocated to crowdworkers: crowdworkers select the point on the slider that yields the best visual appearance. Koyama et al. [87] apply various quality control strategies to address aspects of subjectivity in this assessment. In this chapter, subjectivity in crowdworker input is avoided by directly measuring task completion time: a summative reflection of the perceptual and interactive qualities of the interface. Khajah et al. [77] recruited crowdworkers in their evaluation of Bayesian optimisation to find game parameters that maximise user engagement. As an indicator of engagement, they exploit crowdworker estimates of how much additional (unpaid) time others might spend playing the game. In this chapter, focus is given to the utility and efficiency of the interface and the most closely related performance metric for this purpose is targeted: task completion time.

The unique contribution of this chapter is that a Bayesian optimisation approach is applied to deliver refinement of a diverse set of design features directly based on actual user performance. Through demonstration in three different deployment settings, the potential that this approach has as a design tool with good objectivity, versatility and comparatively low overhead is highlighted.

7.3 Approach

The objective of this chapter is to provide an accessible introduction to, and demonstration of Bayesian optimisation for interface design refinement. Section 7.5 describes the technique

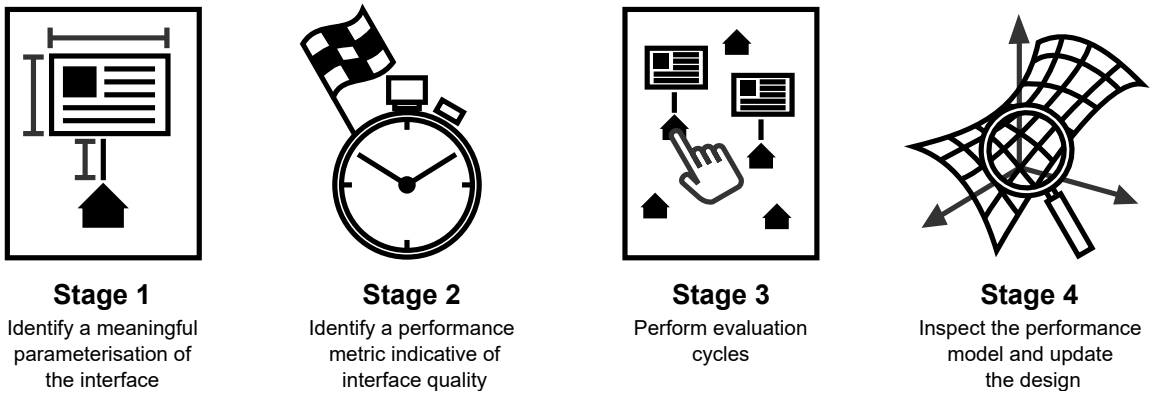


Fig. 7.1 The four stages in the process of designing with Bayesian optimisation.

of Bayesian optimisation contextualised by the interface refinement problem. In contrast to prior work, the formulation presented is readily understood and applied: directly modelling the relationship between interface feature parameters and task completion time. The design process described below is then applied to two illustrative design problems. A further case study serves to show how Bayesian optimisation can be efficiently applied in totally unfamiliar applications and settings. Finally, the broader implications of the findings as they relate to the application of Bayesian optimisation are discussed.

7.4 Designing with Bayesian Optimisation

Bayesian optimisation provides a robust and flexible technique for undertaking objective interface refinement. It is, however, important to provide some structure around the application of this technique in order to deliver meaningful outcomes. To this end, this section offers a high-level description of the process for performing interface feature design using Bayesian optimisation. This process can be divided into four key stages as illustrated in Figure 7.1 and detailed below.

Stage 1. Identify a meaningful parameterisation of the interface

The appearance and behaviour of an interface can be thought of as a product of multiple lower-level design choices. For example, the sizing of textual labels on an interface is one single low-level design choice that ultimately contributes to the appearance and performance of the interface as a whole. Obviously not all low-level design choices are equal in their influence on the resulting performance of the interface, e.g. the choice of font (within reason) may have an effect on the aesthetics of the interface but is unlikely to directly influence performance.

The application of Bayesian optimisation to interface refinement requires the identification of the subset of design choices that are theorised to have the greatest effect on interface performance. This subset of design choices, or *parameters*, represents the *parameterisation* of the interface. Identifying an appropriate parameterisation also involves setting reasonable bounds for a given parameter, i.e. the font size might reasonably be bounded at one extreme by the size that is too small to read and at the other extreme, too large to fit in the display window.

There clearly is a degree of subjectivity involved in the identification of a meaningful parameterisation. This can be alleviated by initial pilot testing and through critical evaluation of the literature and/or related interface implementations.

Stage 2. Identify a performance metric indicative of interface quality

The process of Bayesian optimisation necessitates the measurement of a signal of performance. As will be described later in Section 7.5, a model is constructed to represent the mapping between a design point (i.e. a particular parameterisation of the interface) and its performance as measured through user evaluations. Identifying a robust and clear signal that reflects the quality and performance of the interface is critical to this process. In this chapter, task completion time is the performance metric used.

Stage 3. Perform evaluation cycles

A range of different interface designs (suggested by the Bayesian optimisation technique) are then evaluated. In the context of this chapter, crowdworkers are employed to efficiently perform this evaluation but the same outcomes could be achieved (albeit considerably less efficiently) through lab-based testing.

This evaluation need consider the confounding effects of learning and inter-user variability. In addition, evaluations performed through crowdsourcing need be more vigilant to compliance with task instructions. As described later in Section 7.5.2, the investigation presented in this chapter performed evaluations in batches of users in order to mitigate these confounding effects.

Stage 4. Inspect the performance model and update the design

The Bayesian optimisation approach facilitates the efficient exploration of the design space. After sufficient evaluation, the process must switch from exploration to exploitation: identify the parameterisation that yields 'optimal' performance. Determining when to terminate the evaluation cycles requires some judgement but can be informed by the performance improvement trajectory observed or through subjective user ratings (see Section 7.7.2). Finally, the constructed performance model can then be inspected in order to understand both the optimal setting and sensitivity of design parameters (see Section 7.9.1).

7.5 Bayesian Optimisation

This section provides a brief overview and formulation of the basic principles of Bayesian optimisation. For a detailed explanation and formulation see Snoek [169]. At the expense of completeness, this section provides a simple to understand explanation contextualised by the interface design problem.

Bayesian optimisation works by exploiting a probabilistic model that has been fitted to describe some unknown function. In this study, for example, the goal is to model how users perform when certain interface design features are varied. This function is 'unknown' as there is no way to reliably predict how user performance will be affected by changes to the interface.[1] The conventional approach in Bayesian optimisation is to model the unknown function as a Gaussian process (GP). A GP describes a distribution over functions with the process, $f(\mathbf{x})$, specified by its mean function, $m(\mathbf{x})$, and covariance function, $k(\mathbf{x},\mathbf{x}')$, where

$$m(\mathbf{x}) = \mathbb{E}\big[f(\mathbf{x})\big], \tag{7.1}$$

$$k(\mathbf{x},\mathbf{x}') = \mathbb{E}\big[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))\big]. \tag{7.2}$$

The Gaussian process is then written as

$$f(\mathbf{x}) \sim GP\Big(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}')\Big). \tag{7.3}$$

The $m(\mathbf{x})$ and $k(\mathbf{x},\mathbf{x}')$ are essentially the function parallels of the mean and variance of a random variable. The function, $f(\mathbf{x})$, however, specifies the random variable at location $\mathbf{x}$.

In the interface refinement task, the GP is fitted using data obtained through observations of user performance. Throughout this chapter, task completion time is used as the observation value. An observation instance, representing a particular design configuration of the interface, has parameter values defined by $\mathbf{x}_i$. The crux of Bayesian optimisation is to leverage the GP, fitted to a sequence of observations, $\{\mathbf{x}_{1:t}, \mathbf{f}_{1:t}\}$, to probabilistically determine what new point, $\mathbf{x}_{t+1}$, should be evaluated next. The mean and variance of the Gaussian process predictive posterior distribution after t observations are defined by

$$\mu_t(\mathbf{x}_{t+1}) = \mathbf{k}^T \mathbf{K} \mathbf{f}_{1:t}, \tag{7.4}$$

$$\sigma_t^2(\mathbf{x}_{t+1}) = k(\mathbf{x}_{t+1},\mathbf{x}_{t+1}) - \mathbf{k}^T \mathbf{K} \mathbf{k}, \tag{7.5}$$

[1] This is not to say that certain aspects of user performance cannot be predicted or estimated. Techniques such as KLM and Fitts' Law might allow one to estimate the effect of changes in element sizes or placement. Such techniques, however, struggle when applied to simultaneous variation of multiple interface design parameters with nuanced factor interactions.

where, the Kernel matrix, $\mathbf{K}$ is given by

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_t, \mathbf{x}_1) & \dots & k(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix}, \tag{7.6}$$

and $\mathbf{k}$ is given by

$$\mathbf{k}^T = \begin{bmatrix} k(\mathbf{x}_{t+1}, \mathbf{x}_1) & k(\mathbf{x}_{t+1}, \mathbf{x}_2) & \dots & k(\mathbf{x}_{t+1}, \mathbf{x}_t) \end{bmatrix}. \tag{7.7}$$

As part of fitting observation data to the GP, there are a number of subtle assumptions that must be made about the target function. One of these relates to how closely nearby points in the space are correlated. The covariance between two points is typically referred to as a kernel. The kernel itself has parameters, typically referred to as hyperparameters, which can be thought of as describing the general shape of the function space independent of the data points. There are a range of kernels to choose from with each possessing different properties and expressing different assumptions about the underlying data. Furthermore, different kernels can be combined together to reflect presumed features in the data. An appreciation of the characteristics of the data to be modelled can therefore inform the selection of the most appropriate kernel. For example, the squared exponential (SE) kernel is generally proposed as a good initial selection but assumes a degree of smoothness in the underlying data and therefore may perform poorly in any regions with discontinuities. Duvenaud [40, Chapter 2] and Rasmussen and Williams [148, Chapter 4] provide a helpful overview of a range of kernels and discuss the kernel selection problem. This study employs the automatic relevance determination (ARD) kernel (see [148, pp. 106]) which is a special form of the SE kernel where each input dimension has a dedicated hyperparameter. This gives it the useful property of removing irrelevant input. The ARD kernel is selected for its simplicity and to demonstrate that the approach described can perform effectively even in a rudimentary configuration. The ARD kernel with hyperparameters, $\theta = (\sigma_f^2, \sigma_n^2, l_1, ..., l_D)$ where D is the dimensionality, is defined as

$$k(\mathbf{x_p}, \mathbf{x_q}) = \sigma_f^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_{p_d} - x_{q_d})^2}{2l_d^2} \right) + \sigma_n^2 \delta_{pq}. \tag{7.8}$$

Once the GP is fitted to the observation points, the next step is to determine which new point to sample based on some probabilistic guidance. This guidance comes from an acquisition function: a function that reflects the benefit of sampling a given set of parameter values. The

literature provides many choices for acquisition function. A standard approach based on expected improvement (EI) is used. The EI acquisition function can be thought of as the potential gain that can be obtained, relative to the current best observation, at a given new observation point. This assessment is based on the current model's mean and variance at that point. The EI acquisition function is defined as

$$EI(\mathbf{x}) = \begin{cases} \Big(\mu(\mathbf{x}) - f(\mathbf{x}^+)\Big)\Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \tag{7.9}$$

where $\mathbf{x}^+$ represents the best observed sample in the sample set and

$$Z = \begin{cases} \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})} & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases}. \tag{7.10}$$

Note that the subscripts on μ and σ are omitted for clarity. The concept of the acquisition function is illustrated in a one dimensional example in Figure 7.2. The iterative process of updating the model and selecting the next sample point is summarised in Algorithm 2.

Algorithm 2: Bayesian Optimisation

for $t = 1, 2, ...$ **do**
  Optimise acquisition function, EI, given observation set, $D_{1:t-1}$, to find new sample point: $\mathbf{x}_t = \arg\max_x EI(\mathbf{x}|D_{1:t-1})$
  Sample the objective function: $y_t = f(\mathbf{x}_t) + \varepsilon_t$
  Add new sample to dataset: $D_{1:t} = (D_{1:t-1}, (\mathbf{x}_t, y_t))$
  Update the *GP*.
end

7.5.1 Hyperparameters

As described above, Bayesian optimisation is not completely free from the parameter selection problem. There are several hyperparameters in the ARD kernel which dictate the high level function shape. The signal variance, σ_f^2, is the variance in the signal without noise, i.e. the degree to which the signal varies over the space as a function of the inputs. If there are no observation points in a portion of the space, the standard deviation of the process is σ_f. The noise variance, σ_n^2, reflects the characteristics of the noise added to the underlying signal. As described by Rasmussen and Williams [148, pp. 106], "the $l_1, ..., l_D$ hyperparameters play the role of characteristic length-scales; loosely speaking, how far do you need to move (along a particular axis) in input space for the function values to become uncorrelated."

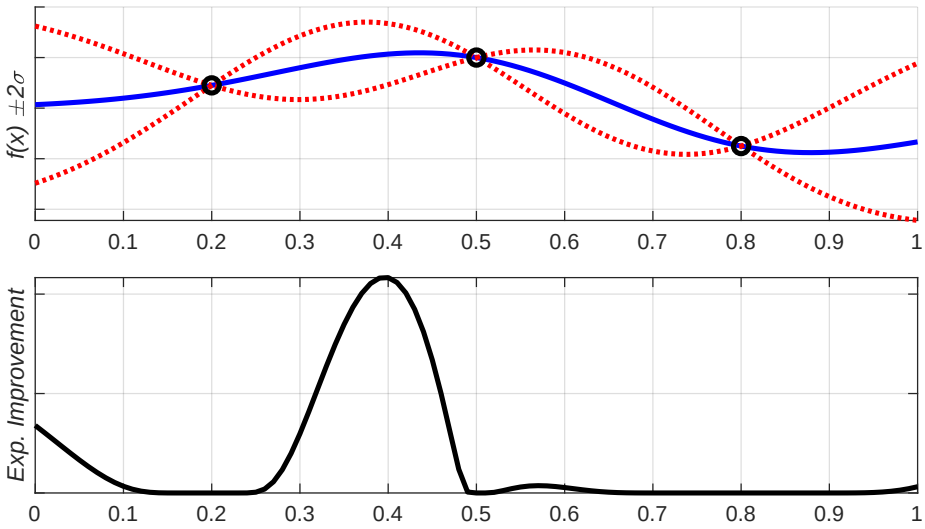


Fig. 7.2 Illustration of Bayesian optimisation in 1D. The top plot shows the Gaussian process approximation of the latent function over the design space. Three observations are shown (black circles) and the uncertainty around these points is visibly reduced. Below is the Expected Improvement over the design space: the potential that a new observation point has to improve upon the current best observation.

Conveniently, there are effective methods for determining appropriate hyperparameter values. One of the main attractions of Gaussian processes for regression models is that the integrals are analytically tractable [148]. As such, it is possible to derive the expression for the marginal likelihood, i.e. the likelihood of the observations given the hyperparameters marginalized over the possible functions. Suitable hyperparameters are found by optimising the marginal likelihood.

7.5.2 Implementation Specific Details

This section documents several details specific to the implementation of Bayesian optimisation used in this study. As suggested by Rasmussen and Williams [148, p. 23], the observation values are rescaled to have zero mean and unit variance. To do this in the absence of prior data, a coarse approximation of the distribution of typical observation values is required. In this study, the observation values are task completion times. Completion time is approximately the product of the number of inspections and time per inspection. Such products approach a log-normal distribution. Based on initial pilot testing and for consistency across the experiments and case study, a mean completion time of approximately 30 s is assumed. To normalise for unit variance, typical task times are estimated to vary between 15 s (half) and 60 s (double) which

corresponds very approximately with a log-normal standard deviation of 0.7. These values could be refined through further pilot testing or using prior task data. The results obtained suggest, however, that coarse approximations yield adequate performance.

A further simplification for implementation purposes is the conversion of the continuous design space into a discrete one. This helps avoid the requirement to exhaustively search the space when optimising the acquisition function. The approach involves evaluating a candidate list of sample points that provide representative coverage of the design space. Appropriate bounds for each parameter are chosen and this sets the limits of the hypercube. The candidate list is then constructed by sampling from the parameter hypercube using a Sobol sequence as described by Snoek [169]. 1000 candidates are sampled in this way. While more candidates provides greater search resolution (at the cost of speed), this value was considered sufficient to demonstrate the approach. Unlike many optimisation problems, there is no expectation that a certain set of parameters will provide universally optimal performance. Far more likely in the case of varied human participants working on different platforms is the identification of an ideal parameter region rather than a distinct peak. Therefore, fine-grained optimisation of the parameters is not necessary. At this point it is also important to highlight a subtle distinction between advantageous exploration and convergence towards a singular 'optimal' design. This chapter uses an optimisation technique but is distinct from pure optimisation. Rather, the expected behaviour under the EI acquisition function is an emerging preference for selection from within a region of good designs (advantageous exploration). At some point, however, this advantageous exploratory behaviour may be overridden by a preference for unexplored regions of high uncertainty exhibiting some potential for improvement.

A further deviation from more typical applications of Bayesian optimisation is the batching approach used. The Bayesian optimisation approach is hypothesised to support the identification of suitable parameter ranges while also reducing imposition on users. To make performance improvement due to parameter refinement testable, prior participant data is incorporated in separate batches. In other words, a batch of tasks with multiple users is completed and this performance data is then fed forward to provide prior information in subsequent batches. Note that this approach is not the same as selecting a set of parameter values to explore (e.g. [52]) as each user is allocated sample points independently of other users within the same batch. Within each batch and for each user, however, the standard process of Bayesian optimisation also incorporates the individual user's prior performance. The hyperparameters are held constant during a batch and then updated based on all data up to and including the most recent batch. In the first batch with no prior data, the hyperparameters were all set to a nominal value (unity). Again, pilot testing or pre-existing data could inform the selection of appropriate values but this chapter demonstrates that the approach can proceed even when naïvely initialised.

7.5.3 Fixed Baseline

A fixed baseline was introduced to serve as a common point of reference across the experimental batches that make up Experiments 1 and 2. Over both experiments, the condition was alternated for each subsequent participant. In the baseline condition, design parameters were uniformly sampled from the design candidate list. For a given participant, this sampling was without replacement such that a participant would not experience the same design twice.

Recall that the candidate list is constructed after first setting sensible bounds on the design parameters. This choice of baseline can therefore be thought of as testing parameters supplied by naïve designers without prior experience or the ability to learn from prior data. This baseline is clearly conservative as even the most naïve designer may be unlikely to choose certain design combinations, even if the individual parameter values are sensible. Nevertheless, this baseline serves as a useful reference point and an important check on population sampling effects.

7.6 Experiment 1: Hotel Search Task

The intent in this study is to evaluate the optimisation approach in the context of a real word interface design problem. As an exploratory venture into this space, a relatively simple task that had good external validity but could still be experimentally controlled was sought. A map search interface, such as encountered on most online hotel booking sites, was chosen. Specifically, this is an interface in which hotel location *pins* are visualised on a map and the user reviews additional details (shown in overlaid *tooltips*) about each hotel by moving the cursor over the map. The task thus requires the user to find a hotel that meets specified criteria.

Given that the actual application is secondary to the demonstration of the approach in this chapter, it is convenient that the map search and review task is generally familiar to users. An assumption is made that the basic interface and interaction learning effects are small and so extensive explanation of the task can be avoided. This means that the variation in the parameters which define the interface design are more likely to be the dominant factor influencing completion time.

The design of a map search interface is a useful demonstration application as it encompasses multiple non-trivial design dimensions. Consider, for example, the timeout period on hiding a tooltip after leaving a pin with the cursor. Setting this value to be too short may prevent the user from making a comparative evaluation while conversely, setting it too long or infinite may cause unnecessary obfuscation of the interface. The timeout period may be chosen by the designer through some self-testing or an informal user study but there is limited objective basis for assuming that value is appropriate for the broader user population. Furthermore, it is easy

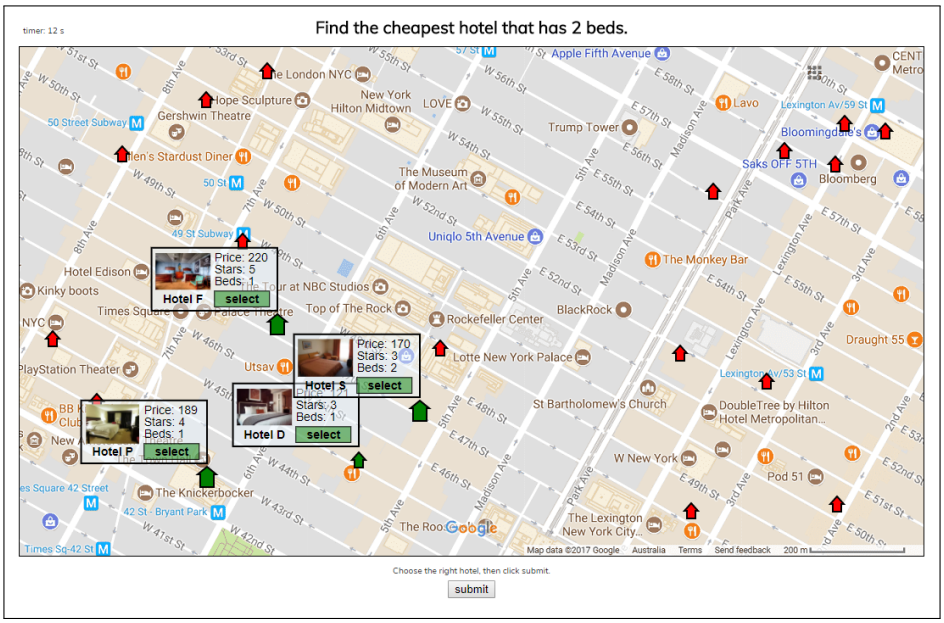


Fig. 7.3 Hotel search task interface. The search criteria are displayed at the top of the interface. The tooltip details for four of the hotels are shown to the bottom left.

to imagine non-trivial interactions between this timeout period and other design parameters such as the distance threshold on initially showing the tooltip.

While the interface design space is obviously theoretically infinite, for practical purposes it is necessary to finitely parameterise the design space. For the purpose of this experiment, the parameterisation of the design space is constrained to five dimensions. Constraining to five dimensions demonstrates utility in a non-trivial parameter selection problem while also maintaining interpretability of the design implications. The five design dimensions chosen are summarised in Table 7.1.

Table 7.1 Interface parameters examined in Experiment 1.

	PARAMETER	BEHAVIOUR
1	*Distance:*	Threshold distance on cursor-to-pin for raising *show tooltip* event.
2	*Delay*:	Timeout before responding to *show tooltip* event.
3	*Decay*:	Timeout before responding to *hide tooltip* event after cursor exits distance threshold.
4	*Size*:	Size of the hotel tooltip.
5	*Opacity*:	Transparency of the hotel tooltip.

Clearly there are many other possible interface design features that could have been chosen. To illustrate the advantages of the presented approach, features that exhibit inherent trade-offs

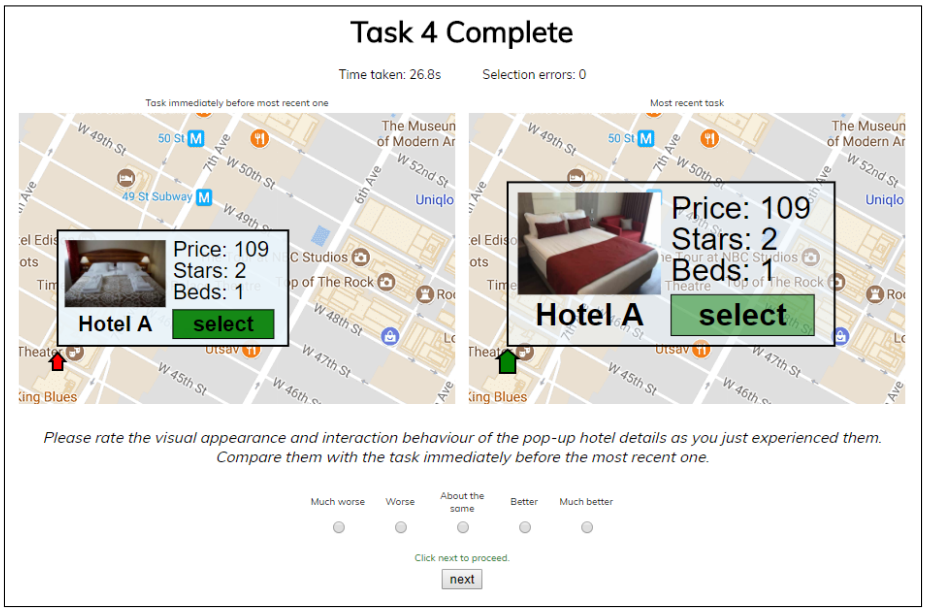


Fig. 7.4 The comparative feature rating page. The left thumbnail shows the previous design configuration while the right shows the most recent configuration.

and expose non-trivial interactions with other features are selected. The map search interface as developed for this experiment is illustrated in Figure 7.3.

Each hotel tooltip lists the name, thumbnail image, price, star rating and bed count. These details were set arbitrarily although effort was made to ensure there was a correlation between star rating and price as well as bed count and price, as per standard hotel room pricing practices. There were always 20 hotels indicated on the map.

7.6.1 Finding Hotels

Participants were instructed to find the hotel on the map meeting specified criteria. For example, the search criteria might say, “Find a hotel that is 3 stars and has 3 beds”. The participant must then search the map and review hotel details until they find the matching hotel. The search criteria were chosen so that there was only one hotel that matched the specified criteria.

Upon finding the matching hotel, the participant must click the *select* button, located on the tooltip, then click the submit button below the map. If an incorrect hotel was selected, the participant is informed of their error and forced to continue their search. A timer recorded the duration of the search task, and the counting timer was displayed on the top left of the map (see Figure 7.3). To avoid circumstances in which the interface parameters are so poor that they prevent the participant from finding the hotel or the participant is otherwise unable to complete the task, the task instance is automatically advanced after 90 s.

After submitting the correct hotel, the participant is presented with a results page. This page lists their completion time and the number of erroneous submissions made. If this was the first search task, no other information was presented and the participant could just click the *next* button to move to the next iteration of the task. If this was the second or later task, the results page would also show two thumbnail maps with a single hotel (see Figure 7.4). These thumbnail maps presented the interface design as per the most recent parameter settings as well as the immediately previous parameter settings. The participant was then asked to rank their experience with the more recent parameter settings on a five point scale: much worse, worse, about the same, better, much better. After assigning their rating, the user could click the *next* button to move to the next task.

A total of 10 search tasks were presented to each participant, each with a different search criteria. The task order was randomly shuffled for each participant but the same 10 tasks were undertaken by all. Each task had a predefined hotel map layout. This layout was randomly generated originally to provide the distribution of hotels on the map but these layouts were then held constant for all users in the experiment described here.

7.6.2 Crowdsourcing Participants

This experiment was formulated as a Human-Intelligence Task (HIT) and participants were recruited through the Amazon Mechanical Turk service. No restrictions were placed on participant qualifications so any Mechanical Turk user, or *worker*, was able to complete the HIT. Workers were limited to completing the HIT only once so all participants in this experiment are unique.

Recall that the Bayesian optimisation procedure demonstrated in this chapter was applied in batches. Batch size was set to 20 participants. At 10 tasks per participant there were 200 unique parameter observations per batch. The procedure was executed for five batches in both the baseline and Bayesian optimisation condition. Therefore, there were 200 unique participants in the experiment. This procedure is illustrated in Figure 7.5.

Participants were compensated US$1 for their participation. The HIT, including reading the introductory material and instructions, took approximately 10 minutes to complete. After completing all tasks, participants could elect to provide basic demographic information. In total, 79 specified female, 118 male, and three participants did not respond. Participant ages ranged from 21 to 65 with a median of 30.

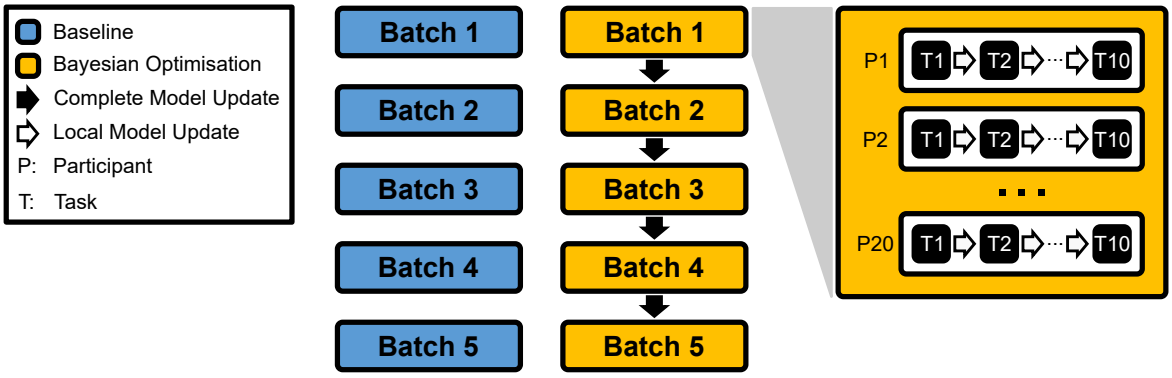


Fig. 7.5 Illustration of the batching and model update procedure for the baseline and Bayesian optimisation conditions. As described in Section 7.5.2, in the Bayesian optimisation condition all task observations collected up to and including the current batch are used to update the complete model for the next batch. Within a given batch only the individual's prior task performance is used to further update their specific local model.

7.6.3 Performance Results

The batch results are summarised in Table 7.2 and Figure 7.6. Note that automatic advances of the task (i.e. where the task was not completed within 90 s) are excluded from these results although they are included as observations in the Bayesian optimisation step. Note that discrete task time represents the time to complete a single search task and that each worker was presented with 10 search tasks. Each batch contained 20 workers so the n value reported in Table 7.2 indicates how many of the 200 tasks were actually completed.

The results as obtained in chronological order of completion are now reviewed. In Batch 1, the boxplots of the baseline and Bayesian optimisation conditions indicate very similar performance levels. A two-sample t-test on the log times reveals no significant difference between the samples (p=0.88).

Table 7.2 Median task times and completion counts in Experiment 1.

	Median Task Time (s) [n]	
Batch	*Baseline*	*BO*
B1	32.9 [165]	30.8 [166]
B2	34.0 [175]	21.5 [170]
B3	31.1 [161]	20.5 [187]
B4	34.1 [172]	21.3 [183]
B5	29.6 [165]	26.1 [191]

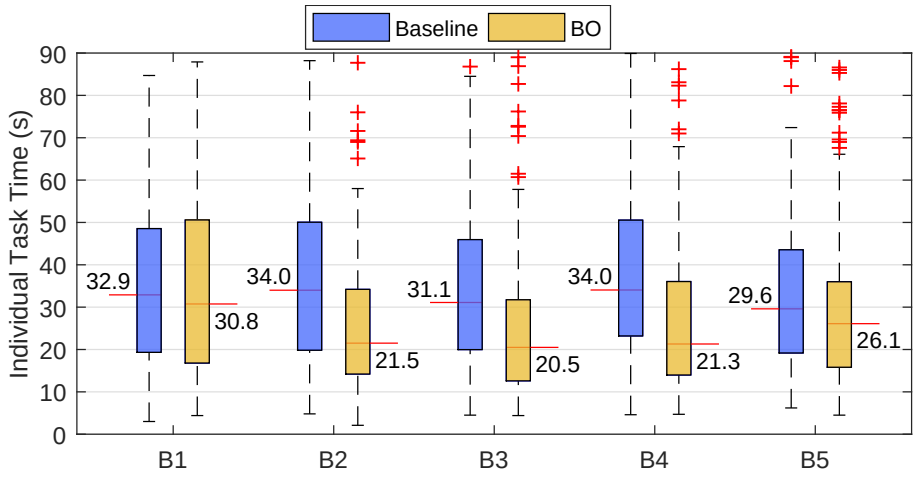


Fig. 7.6 Boxplots of task completion time over the five batches for both conditions in Experiment 1. The red crosses indicate outliers based on $Q_3 + 1.5 \times (Q_3 - Q_1)$.

This result is intuitive given that at this stage, the Bayesian optimisation procedure has limited data upon which to model the parameter space. Given the parameter space is $\mathbb{R}^5$ there are insufficient samples to cover the corners of the hypercube. With insufficient data to make any firm assumptions about the space, the acquisition function typically encourages sampling that covers the space as broadly as possible.

Batch 2 yields an improvement both relative to Batch 1 and its paired Baseline condition. The median completion time of 21.5 s represents a 30% reduction in median completion time compared to the same condition in Batch 1. A two-sample t-test reveals a significant difference between the Bayesian optimisation and Baseline conditions ($p < 0.01$). This result suggests that the prior data incorporated from Batch 1 has encouraged the exploration of regions of the parameter space where there are actual performance improvements to be obtained. Batch 3 extends this further but with reduced gains. The median task time of 20.5 s is the lowest achieved and represents a 33% reduction relative to Batch 1.

It is interesting to note that Batches 4 and 5 remain significantly faster than the Baseline but are slightly elevated compared with the peak obtained in Batch 3. The interface improvements derived through the Bayesian optimisation procedure are also evident in the increased task completion rates (n) (see Table 7.2), peaking at 95.5% in Batch 5 compared with 83% in Batch 1.

The performance plateau and subsequent increase in completion times can be explained by further exploration of the design space. As more observations are made in the region of good performance, this reduces the uncertainty in that region. The acquisition function used will always seek to maximise the expected improvement. At some point it is possible that although the predicted mean for a largely unvisited region is poor, the uncertainty in

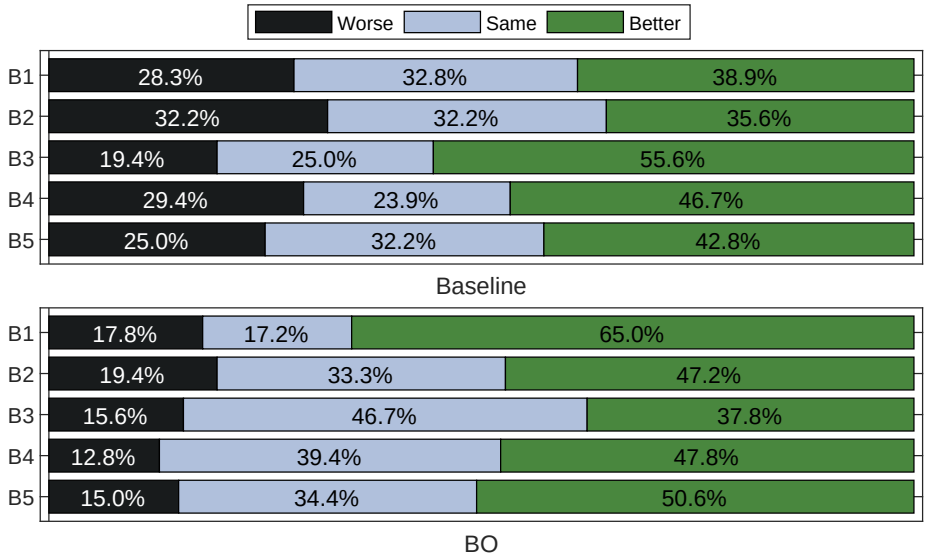


Fig. 7.7 Proportion of interfaces rated by users as Worse, Same or Better over the 5 batches in Experiment 1. The Baseline condition is shown at the top and the Bayesian optimisation condition below. The proportion of Same user ratings steadily increases over batches 1-3 in the Bayesian optimisation condition.

this region might promote its investigation. Although ultimately useful for modelling the complete design space, such explorations will manifest as poor batch aggregate performance. This problem is typically referred to as the exploration versus exploitation trade-off. At some point, it is better to 'exploit' the known region of good performance by taking finer and finer observations from within that region. As described in Section 7.5, a coarse candidate list was hypothesised to be appropriate for such interface refinement problems as fine parameter variation may not necessarily translate into noticeable difference in the interface. Nevertheless, there are alternative acquisition functions and formulations in the literature that better manage this transition between exploration and exploitation (see, for example, Lizotte [107]).

7.6.4 Interface Variation Ratings

An alternative perspective on the interface feature refinement procedure is provided by looking at the participant ratings made after each task (except for the first task where no relative comparison is possible). Figure 7.7 presents the rating proportions grouped based on three categories: Same (representing 'about the same' on the rating scale), Better (representing 'better' and 'much better') and Worse (representing 'worse' and 'much worse'). This reduction is done to improve the clarity of the observable trends.

As the refinement process proceeds in the Bayesian optimisation condition, it is expected that the range of plausible parameter settings that offer potential improvements narrows. This trend is observable in Figure 7.7 where the Same counts steadily increase over batches 1-3. This largely comes at a cost of a smaller proportion of perceived improvements. The final two batches maintain a positive improvement bias but as observed in the median completion data, this does not translate into distinct performance improvement.

A further interesting result visible in Figure 7.7 is the high degree of variability for the Baseline condition. There is no reason that participants should perceive a task-to-task improvement in the interface in the Baseline condition yet there remains a positive bias over the batches. This observation may be due to a recency effect bias, but does suggest that when unconnected to a known model driving change such comparative preference data may be unreliable.

Fig. 7.8 The VR hotel search task in Experiment 2. The mobile device presents a window into the virtual world. The user controls the view of the scene by adjusting the orientation of the device. The gaze cursor (shown in orange) is used to inspect hotel information and locate the hotel that matches the specified criteria.

7.7 Experiment 2: Mobile VR Search Task

The results from Experiment 1 suggest that Bayesian optimisation presents a viable approach for refining 2D interface design parameters. As a subsequent test of the versatility of the procedure, a less familiar and arguably more challenging interface design problem is investigated.

The 2D hotel search task was adapted to run as a mobile based quasi-virtual reality application. Rather than presenting the hotels on a 2D map surface, 3D hotel icons were displayed on an inclined map plane inside a rudimentary virtual environment. A screenshot of the task environment is presented in Figure 7.8. The view of the virtual environment is adjusted by using the mobile device as a window into the virtual world. It is important not to misconstrue

the investigation of this particular interface as a suggestion for its practical use in a real-world hotel booking application. Rather, it serves as a demonstration of the Bayesian optimisation approach in a more challenging interface deployment setting but with common design features.

For consistency, this task evaluated the same parameterisation of the interface used in Experiment 1 with some minor adjustment for the differing interaction behaviour (see Table 7.3).

Table 7.3 Interface parameters examined in Experiment 2.

PARAMETER		BEHAVIOUR
1	*Distance:*	Threshold distance on projected view-centre to pin for raising *show tooltip* event.
2	*Delay*:	Timeout before responding to *show tooltip* event.
3	*Decay*:	Timeout before responding to *hide tooltip* event after cursor exits distance threshold.
4	*Size*:	Size of the hotel tooltip.
5	*Opacity*:	Transparency of the hotel tooltip.

As in Experiment 1, the participant must find the hotel that matches the specified criteria. Five batches were executed in both the Baseline and Bayesian optimisation conditions. Batch size per condition was 20 participants as in Experiment 1. Participants could only complete the experiment once so all participants are unique. Note that participants who completed Experiment 1 were not prevented from completing Experiment 2. Of the 200 participants, 100 specified female, 97 male, one other, and two participants did not respond. Ages ranged from 18 to 64 with a median of 29. Participants were compensated US$1.20 for completing the HIT.

After each task, participants were again presented with the performance summary and rating page. Due to the confined screen space available in the mobile setting, no thumbnail reminders of the interface features were presented.

7.7.1 Performance Results

The results from Experiment 2 are summarised in Figure 7.9 and Table 7.4. The distribution of completion times in Batch 1 is broadly consistent between the Baseline and Bayesian optimisation conditions. In Batch 2 there is an observable reduction in median completion time in the Bayesian optimisation condition. There are further, but more marginal reductions in Batches 3 and 4. As in Experiment 1, there is a subsequent elevation in completion times in Batch 5. Again this is likely a consequence of disadvantageous exploration.

The difference in median completion time between Batch 1 and Batch 4 in the Bayesian optimisation condition represents a reduction of 24.8%. The difference between conditions in

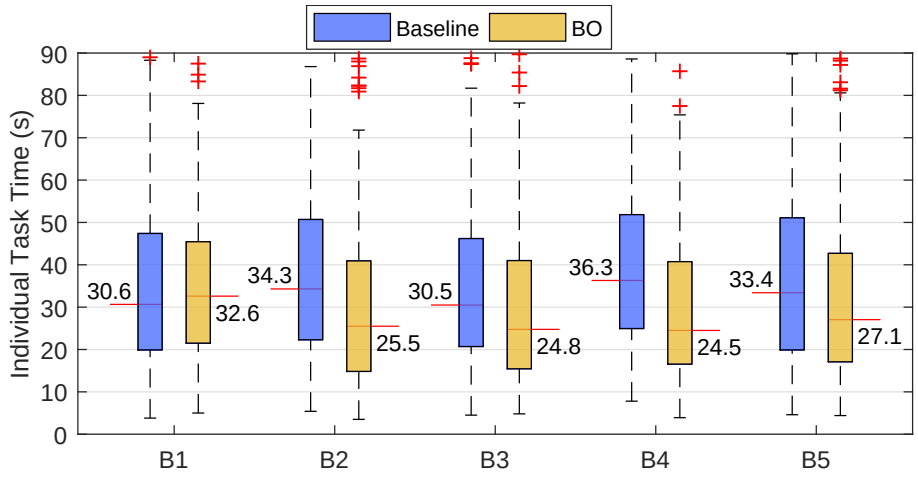


Fig. 7.9 Boxplots of task completion time over the five batches for both conditions in Experiment 2. The red crosses indicate outliers based on $Q_3 + 1.5 \times (Q_3 - Q_1)$.

all but the first batch ($p = 0.31$) are significant ($p < 0.01$) based on two-sample t-tests applied to the log time.

7.7.2 Interface Variation Ratings

The post-task interface ratings are summarised in Figure 7.10, grouped into 'Worse', 'Same' and 'Better'. A distinct deviation from the results of Experiment 1 is the negative bias visible in the Bayesian optimisation condition for Batch 1. Interestingly, this bias is reversed through batches 2 to 4.

Recall that, due to the limited screen space in the mobile setting, no thumbnail interface was presented to help participants recall the recent interface designs. It is reasonable to expect that this would make participant comparative judgements even more subjective and prone to error.

Table 7.4 Median task times and completion counts in Experiment 2.

	Median Task Time (s) [n]	
Batch	*Baseline*	*BO*
B1	30.7 [170]	32.6 [164]
B2	34.3 [182]	25.5 [187]
B3	30.5 [182]	24.8 [192]
B4	36.3 [180]	24.5 [188]
B5	33.4 [181]	27.1 [186]

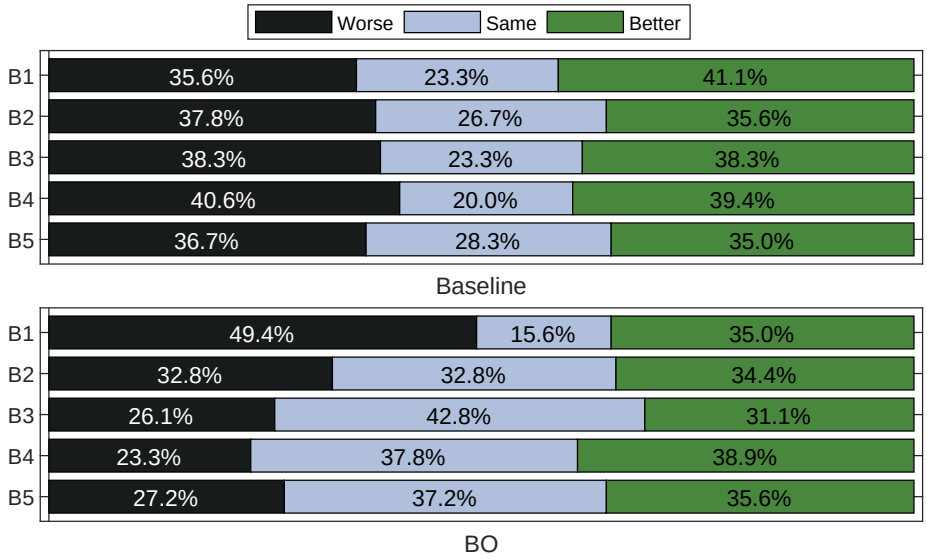


Fig. 7.10 Proportion of interfaces rated by users as Worse, Same or Better over the 5 batches in Experiment 2. The Baseline condition is shown at the top and the Bayesian optimisation condition below. As in Figure 7.7, the proportion of Same user ratings steadily increases over batches 1-3 in the Bayesian optimisation condition.

A consistent feature visible in both Figure 7.7 and Figure 7.10 is the peaking of 'Same' judgements in Batch 3. In both Experiments, Batch 3 is the batch where 'Same' ratings become the majority category. In both Experiments, Batch 3 is also the batch by which the most significant performance improvements have already been achieved.

7.8 Design Case Study: Mobile AR Task

Experiments 1 and 2 highlight the power of Bayesian optimisation in delivering refinements to the interface in an objective and probabilistic fashion. As a further test, the refinement approach is applied to a radically different and unfamiliar design problem. Furthermore, the requirement to capture the Baseline condition is removed. This fully enables the efficient parallelisation of the technique through crowdsourcing. Experiments 1 and 2 serialised the participants in order to ensure strictly alternating test conditions. Free from this constraint, it is theoretically feasible to launch a full batch for crowdworkers to complete in parallel.

The novel interface design challenge tackled is the refinement of an interactive through-the-screen augmented reality experience. There is very limited research providing guidance on the design of interactive through-the-screen AR, particularly when deployed as a web application.

A design challenge particularly relevant to mobile device AR is gaze cueing. More specifically, exploiting the placement and behaviour of virtual content to encourage users to look

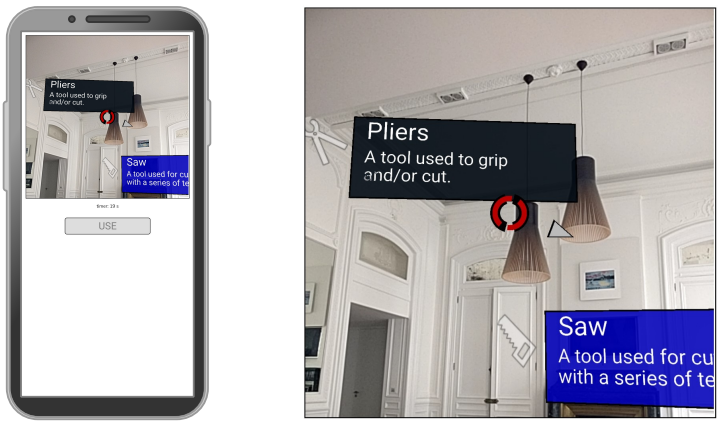


Fig. 7.11 AR interface (left) and detail of scene (right). The user looks around their local environment to locate a series of virtual objects (styled as virtual tools). After all objects are located, the user is instructed to find and select a specified tool.

at certain target objects in the scene (whether physical or virtual). A simple web application was developed that constructed an AR experience in which users must review items in the scene then locate a specified item. For the purpose of the case study, this was framed as a task involving an inventory of virtual tools overlaid on the physical environment. To complete the task, the participant must sequentially find and review all tools in the scene. After all tools were reviewed, an instruction would appear to find a specific tool. Figure 7.11 shows a screenshot of the tool finding AR interface.

The AR experience was constructed using the device camera feed as the background canvas for the virtual scene[2]. To promote a more contextually connected experience, the colouration of each tool description panel would adapt to the physical background. As a rudimentary strategy, the description panel colour was set based on the 180° offset from the mean hue of the background immediately behind the description panel. In addition, the text colour would correspondingly switch between black or white depending on the perceived brightness of the description panel in order to promote readability [150].

This interface was parameterised into five design features. Some of these features are familiar with intuitive implications for task performance while others are highly novel with unpredictable influences. The bounds on parameters were set based on preliminary self-testing among the researchers involved in the study. Each of the design features is summarised in Table 7.5.

In Experiments 1 and 2, limited subsequent benefit after two batches was observed. The batch in which participant ratings of ‘Same’ became the dominant rating also appeared to pro-

[2]Built using A-Frame <https://aframe.io/>.

Table 7.5 Interface parameters examined in the Design Case Study.

	PARAMETER	BEHAVIOUR
1	*Background Timeout:*	Focus time required to mark tool as visited.
2	*Foreground Timeout*:	Focus time required to return tool to foreground (required to select a specified tool).
3	*Lightness Offset*:	Offset applied to panel colour to discriminate between the in-focus and backgrounded state.
4	*Gaze Guidance Grouping*:	Threshold on grouping the guidance arrows towards unvisited tools.
5	*Opacity*:	Transparency of the description panel.

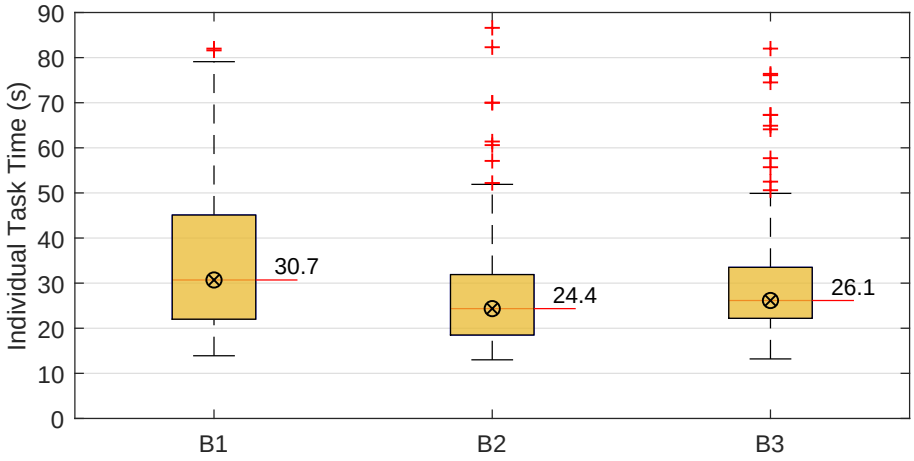


Fig. 7.12 Boxplots of task completion time over the three batches run in the mobile AR task design case study. The red crosses indicate outliers based on $Q_3 + 1.5 \times (Q_3 - Q_1)$.

vide a reliable indication of the point of limited subsequent improvement potential. Therefore, this case study uses this indication as the trigger for terminating the refinement process.

7.8.1 Results

Based on the participant rating trigger proposed, the observation that the 'Same' category became the majority rating in Batch 3 suggested that the refinement procedure be concluded. Between Batch 1 and 2, the median completion time was reduced by 20.7%. There is then a marginal elevation in completion time between Batch 2 and 3. The plateauing of performance is reached earlier than in Experiments 1 and 2 but demonstrates that the proportion of 'Same' ratings provides an informative marker.

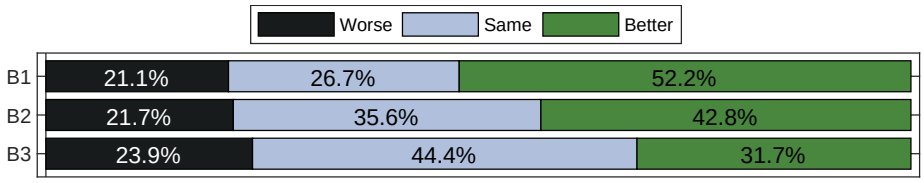


Fig. 7.13 Comparative feature rating proportions in each batch for the AR case study. The increasing proportion of Same user ratings over batches 1-3 is consistent with Figures 7.7 and 7.10

In total 60 participants completed the task (28 female, 30 male, 2 unspecified). Participant ages ranged from 19 to 46 with a median of 30.5. Participants were compensated US$1.20 for completing the HIT. The case study results are summarised in Figure 7.12. The median task completion times and completion counts [n] over Batches 1 to 3 were: 30.7 s [178], 24.4 s [182], and 26.2 s [156] respectively. Figure 7.13 plots the participant post-task rating proportions.

7.9 Discussion

The results of this study highlight the value of Bayesian optimisation as a method for supporting interface design through online user testing. Substantial reductions in task completion times were observed in all three applications of the approach. The method is also able to accommodate the high levels of noise introduced by inter-user performance variability, inter-task variability and task learning effects. Under the rudimentary configuration of Bayesian optimisation applied, the simple method for triggering termination based on perceived interface changes may be sufficient. More advanced configurations are available in the literature which help to better transition between exploration and exploitation.

A limitation of this work is that it is difficult to distinguish between truly optimising design parameters and merely eliminating poorly performing regions of the design space. Pruning bad regions of the design space would yield the same result in terms of reduction in median task completion time. Future work will investigate whether this is the case but the ability to robustly reject bad designs may in itself be useful.

The approach also has other practical advantages that may help streamline interface refinement exercises. Compared with alternative methods for evaluating the complete design space, Bayesian optimisation can help to ensure that subsequent batches of the participant group benefit from the efforts of the previous group. Therefore, a well-functioning optimisation process of reasonable dimensionality should typically have worst case performance in the first batch. This may be useful in predicting task time or adjusting pay scales as the task becomes

faster and easier to complete. It is likely that the variance in inter-user experience is also reduced.

There are several open questions that will be addressed in future work. First, in this study optimisation is performed based on performance metrics only. It would be informative to investigate how performance metrics might be complemented by pair-wise user ratings such as those collected. Second, this data driven approach might successfully integrate theory-driven design methods such as those described by Micallef et al. [121]. In particular, such approaches might provide structured methods for selecting which parameters to refine and appropriate bounds. They may also be helpful in determining appropriate candidate resolution by reference to just noticeable differences. Third, the investigation is constrained to relatively low dimensionality so as to provide a simple demonstration of the method. The scalability of Bayesian optimisation has received some attention in the machine learning community (see e.g. [191, 21]) but the implications of and procedures for dealing with a high dimensional design space remains as future work.

7.9.1 Querying the Design Model

An ancillary benefit of Bayesian optimisation for interface feature refinement is that the procedure yields a model that has other potential uses. As an example, the model can be queried to examine the sensitivity around the optimal design candidate identified in Experiment 1. The GP model incorporates all the collected samples and reflects the relationship between the design parameters and task completion time. It inherently accommodates and reflects the uncertainty in the sampling process. Figure 7.14 plots the variation in estimated task time as the parameter values are varied, one-at-a-time. This plot indicates that it may be possible to eke out further improvements by minor parameter tweaks. The plot also suggests that the *Decay* and *Size* parameters have the dominant effect on task time. Figure 7.15 provides an alternative perspective on this estimate of the relationship between design parameters and performance by also plotting all collected observations. Note that this plot shows the 5D candidate samples collapsed into just their *Decay* and *Size* parameter components and so in not strictly an accurate representation of the model. Nevertheless, it highlights the degree of observation noise inherent in sampling user task performance.

The generated performance model may also be used for simulation. The effect of proposed design changes may be estimated, not only to determine an approximate delta but also to estimate the anticipated distribution of performance. An extension of this idea is the ability to use the same model to identify parameters that minimise performance variation despite elevated average performance. This may indeed be a preferred outcome in some applications and use cases.

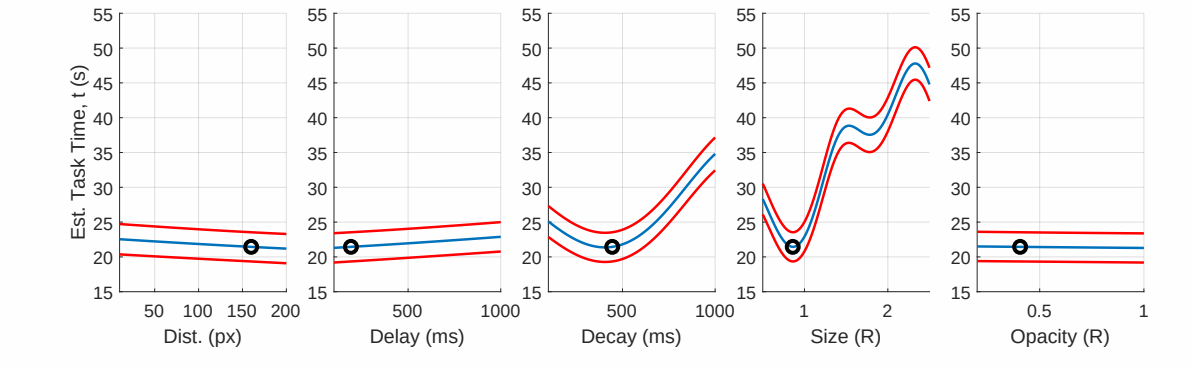


Fig. 7.14 Sensitivity around optimal design candidate (black circle) as indicated by the mean (blue line) and $\pm 2\sigma$ (red lines) of the GP. Note that this is the latent function model prediction which does not reflect the additive signal noise.

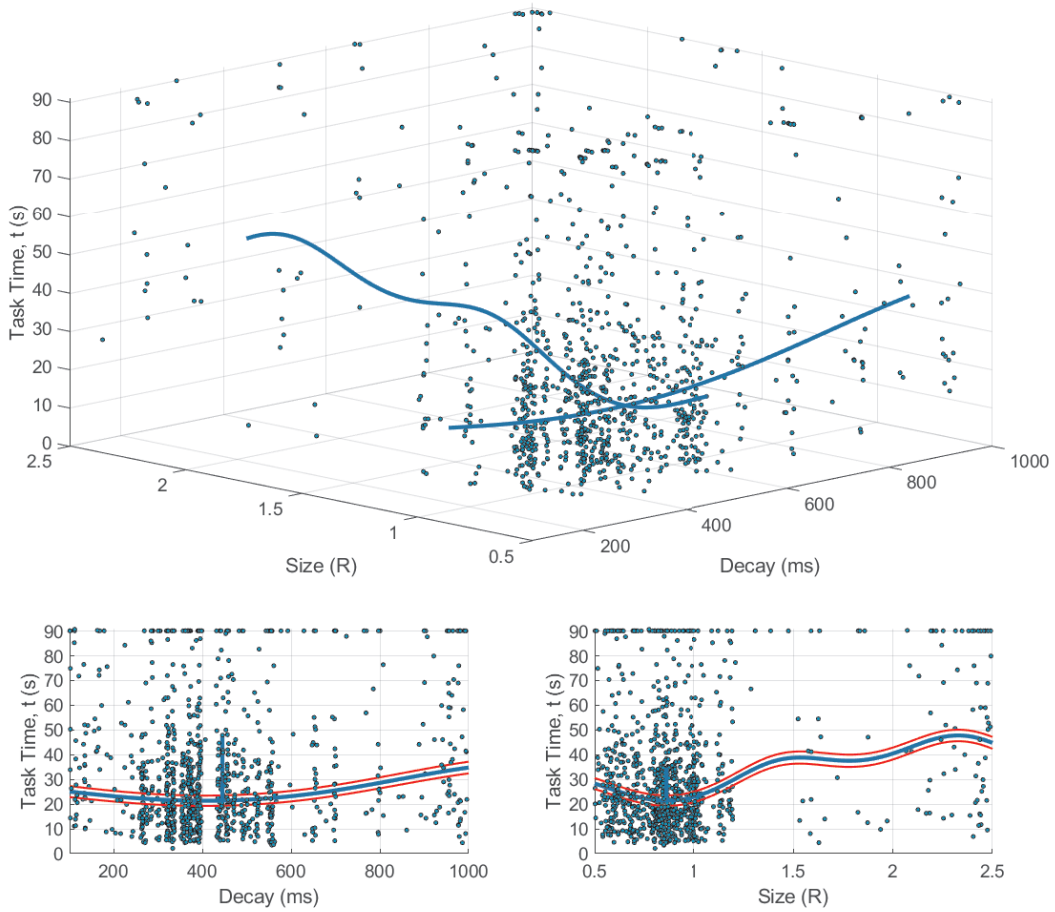


Fig. 7.15 Illustration of the *Decay* and *Size* parameter sensitivity overlaid with all collected observations. Note that the 5D sample points are collapsed into just their *Decay* and *Size* parameter components and so the plot is not strictly an accurate representation of the model. As in Figure 7.14, plot shows mean (blue line) and $\pm 2\sigma$ (red lines) of the GP.

7.10 Conclusions

Bayesian optimisation offers a powerful tool to support the objective refinement of interface designs. This has high potential value to designers given the low overhead of the approach and the fact that there is no subjective tuning required. The only real input required to initialise the process in the example presented is the setting of the bounds on the parameter values. A batched approach to incorporating prior user performance data is shown to deliver clearly detectable improvement in the interface with reductions in aggregate task completion time of between 33.3% and 20.7% in the deployments tested. These results indicate that there is significant potential in this method as a generic means of supporting designers in objective and data driven refinement of their interfaces.

7.11 Research Question 4 and the Design Process

This chapter has sought to address *Research Question 4: How can the unfamiliar and high dimensional design space for mixed reality applications be efficiently explored and refined through probabilistic optimisation?* As previously discussed, this research question reflects the challenging situation facing MR application designers who are without established guidelines and prior experience. The potential of Bayesian optimisation as a probabilistic optimisation approach suited to novel and unfamiliar interface design problems has been highlighted.

The examination of this research question has required a more process focused perspective than the system focused perspectives of Chapters 4, 5 and 6. Nevertheless, there is consistency in the application of computational and probabilistic techniques to develop an understanding of the user and, through this, improve the interface design. There is also correspondence between Stages 3 and 4 of the emergent design process described in Section 2.3. Here, the sensitivity of the design parameters and their refinement are performed as an integrated part of the user evaluation.

Chapter 8

Conclusions

This thesis offers a detailed investigation of probabilistic user interface design for augmented and virtual reality. This chapter summarises the core contributions of the thesis, limitations of the work and worthy avenues of future research. The research questions outlined in Chapter 1 are revisited and examined with respect to the outcomes presented in the previous chapters.

8.1 Research Question 1: Characterisation

Research Question 1: How can a designer obtain an understanding of the probabilistic characteristics of an interface; and, how can this understanding inform design in mixed reality?

Chapter 4 demonstrates the process of isolating and describing the probabilistic characteristics of an interface as a precursor to detailed design. The case study investigates two fundamental design choices for supporting text entry in VR. The characterisation of this interface is performed with low developmental effort by employing a simulated decoder and otherwise fairly rudimentary interaction strategies, answering the first part of *Research Question 1*. The high level performance envelopes and low level micro metrics identified contribute knowledge that is readily actionable in the subsequent development of a more complete probabilistic text entry method. Section 4.6 addresses this second part of *Research Question 1*.

The key research contributions of this investigation are:

1. *An empirical investigation isolating the influence of two key design decisions for text entry in VR: finger engagement and physical surface alignment.*

   The results presented in Section 4.5.1 demonstrate that surface alignment is a significant factor in supporting high entry rates. The use of 10 fingers was found to provide no entry rate improvement over 2 fingers and was generally associated with higher error

rates. Notably, these two factors were investigated without the need to develop a fully functional probabilistic decoder. The data pertaining to this empirical study is publicly available on the university repository at <https://doi.org/10.17863/CAM.41547>.

2. *An awareness of the underlying behavioural factors dictating performance and potential strategies for targeting these factors to improve performance.*

   The range of micro metrics examined in Section 4.5.2 highlight important performance and behavioural differences between input modes. For example, the press reversal distance results are directly informative of tracking system requirements. The awareness of accuracy and error types over the layout, as represented by Figures 4.8 and 4.9, can likewise be encoded into an interaction model that estimates the likelihood of a key given the touch location as well as the likelihood of it being a spurious touch.

8.2 Research Question 2: Adaptation

Research Question 2: How can a data-driven probabilistic preference model for the appearance of virtual content in mixed reality be efficiently obtained; and, how can this be leveraged to enable adaptation of mixed reality applications to uncertain deployment contexts?

Chapter 5 introduces a method for performing efficient data collection of AR deployment contexts through crowdsourcing and a low-fidelity mobile AR experience. This data collection method illustrates how a probabilistic preference model relating background context to virtual content appearance can be efficiently derived, addressing part one of *Research Question 2*.

Section 5.8 explores one strategy for leveraging the probabilistic qualities of this model to deliver dynamic contextually-adaptive text content in AR. The preference distributions identified in the collected dataset for colouration and placement were encoded into a procedure for estimating the preferred placement and colouration of text panels given the background setting. This procedure serves as an example of how the preference model can be leveraged, addressing the second part of *Research Question 2*.

The key research contributions of this investigation are:

1. *A method for conducting AR experiments in the user's own context via crowdsourcing.*

   The method described in Section 5.4 prompts users to capture diverse images of their environment. In two experiments leveraging the efficiency and reach of crowdsourcing, almost 2000 images were collected by 400 users from 22 different countries.

2. *A protocol for mitigating the privacy concerns of crowdworkers as they share images of their local contexts.*

   The privacy protocol embedded in the AR experimental method was found to be effective at addressing the privacy concerns of crowdworkers. The results presented in Section 5.6.1 indicate an approval rate in the image capture task of 99.5%. Interestingly, a majority of the responses to the privacy survey undertaken as part of Experiment 1 indicated limited privacy concerns regarding sharing images but that the provision of a review and obfuscation protocol was important and useful (see Figure 5.10).

3. *A demonstration of the method in building a probabilistic preference model to enable dynamic adaptation of virtual content given background context in AR.*

   The procedure for dynamically adapting textual labels to background context is described and evaluated in Section 5.8. In a preliminary user study assessing label text scanning and response times (as a proxy for legibility), the dynamic adaptation procedure is shown to be as performant as a naïve placement and colouration strategy while providing the additional robustness inherently afforded by the method.

8.3 Research Question 3: Inference

Research Question 3: How can probabilistic inference be exploited to accommodate high levels of input noise in mixed reality applications to deliver more efficient interactions?

Chapter 6 investigates the benefits of inference in accommodating noisy input in AR. The VISAR text entry method leverages a probabilistic decoder to disambiguate input made uncertain by sensor and articulation imprecision. The decoder makes it feasible to build an input strategy based on direct touch by virtualising the input surface. The VISAR keyboard is shown to deliver enhanced user performance against a gaze-based non-probabilistic baseline. In addressing *Research Question 3*, Chapter 6 shows that inference methods can not only help mitigate noisy input under normal circumstances but also deliver ancillary benefits such as enabling typing in low-occlusion display configurations.

The key contributions of this investigation are:

1. *Six design principles informing the development of text entry methods for AR HMDs.*

   The design principles for text entry methods in AR presented in Section 6.4 are derived through the investigative process described in Section 3.2. These are: *DP 1* rapid input

selection; *DP 2* tolerance to inaccurate selection; *DP 3* minimal occlusion of field-of-view; *DP 4* intelligent word predictions; *DP 5* fluid regulation between input modes; and *DP 6* walk-up usability and acceptance. The iterative stages of design and evaluation summarised in Section 6.5.4 provide confidence that these principles accurately reflect the key factors influencing user experience and performance in AR text entry.

2. *A novel keyboard implementation based on inference employing an error-tolerant touch-driven paradigm.*

   The VISAR keyboard implementation described in Chapter 6 is guided by the six design principles identified. It leverages a familiar touch-based interaction strategy that is tolerant to errors, provides fluid switching between input activities and allows users to type effectively with minimal visual features.

3. *An empirical investigation comparing the novel keyboard implementation with an established non-probabilistic gaze-then-gesture baseline.*

   As described in Section 6.9.4, the performance of the final design iteration of the VISAR keyboard exceeds that of the state-of-the-art baseline by 19.6%. This evaluation serves to validate the identified design principles as well as the broader probabilistic user interface design approach. The data pertaining to this and other evaluations presented in the chapter is publicly available on the university repository at <https://doi.org/10.17863/CAM.25391>.

8.4 Research Question 4: Probabilistic Optimisation

Research Question 4: How can the unfamiliar and high dimensional design space for mixed reality applications be efficiently explored and refined through probabilistic optimisation?

Chapter 7 demonstrates how making interface design decisions can be supported by probabilistic optimisation. The case study examines interface design challenges of increasing complexity from a conventional 2D interface through to a novel mobile AR interface. Crowdworkers are employed to efficiently explore and refine the design space using Bayesian optimisation. Chapter 7 demonstrates how Bayesian optimisation can deliver an objective means for making design choices in a high-dimensional design context, addressing *Research Question 4*.

The key research contributions of this investigation are:

1. *An evaluation of Bayesian optimisation for interface design refinement of two challenging design spaces.*

The potential of Bayesian optimisation in assisting interface design refinement is demonstrated in two experiments presented in Sections 7.6 and 7.7. In both experiments, the Bayesian optimisation approach achieved a substantial reduction in task completion time of between 20-30% relative to the fixed baseline. The data pertaining to these experiments is publicly available on the university repository at <https://doi.org/10.17863/CAM.34781>.

2. *Demonstration of the approach in a novel AR design case study.*

   As an illustration of the applicability of the approach to novel and unfamiliar design problems with high dimensionality, Section 7.8 describes an evaluation of a web-based AR interface. A 20% reduction in task completion time is observed over the course of the refinement procedure.

3. *Implementation guidance for crowdsourcing interface design refinement using Bayesian optimisation.*

   A high-level description of the process for performing interface feature design with Bayesian optimisation is presented in Section 7.4. In addition, the potential for subjective user ratings to provide termination guidance is investigated and discussed as is the information contained in the learnt model regarding performance sensitivity to design parameters.

8.5 Limitations

The primary limitation of this thesis is that it is difficult to demonstrate the advantages of probabilistic user interface design in relative terms. There is no clear baseline against which many of the methods can be evaluated. The premise upon which this thesis is argued is that a structured approach is likely to deliver better or at least more consistent outcomes than an unstructured alternative.

Limitations specific to the four case studies explored are discussed within the corresponding chapter. Other more general limitations of this work are summarised and discussed below. Where possible, suggested remedies for these limitations are proposed.

- **Choose the right tool for the job.** The additional complexity introduced by a probabilistic approach may not always be warranted when a simpler solution may do. As formal guidance and experience in MR application design develops, it may be possible to extrapolate this guidance and experience to related design problems without the need

for a probabilistic approach. This tension is briefly discussed in Section 2.3 in terms of identifying the 'need' as a precursor to solving the more specific design problem.

- **External validity of data.** The quality of any probabilistic characterisation or model is inevitably dependent on the quality of the underlying data. Unavoidably, the bulk of the data used in this thesis was collected through controlled user experiments in contrast to actual in-use data. This raises potential concerns about the external validity of the data for a practically deployed application. As discussed in Section 4.7.1, for example, the inability to perform error corrections and the constraints placed on posture mean that the characterisation of typing in VR is potentially reductive and incomplete. While this is a reasonable concern, in practice access to in-use data could subsequently be incorporated through the same techniques to address any validity concerns.

- **Case study selection.** As described in Section 3.5, the choice of case studies was motivated by demonstrating both a variety of techniques and application domains. While this objective was achieved, the examination of several use cases from largely orthogonal perspectives limited the transfer of insight between case studies. Nevertheless, a single monolithic use case examined incrementally may have raised other concerns about the generality of the approach.

- **Hardware and device agnosticism.** This research project has drawn an artificial boundary to exclude the lower-level device and sensor characteristics from the examination of the user interface. This is justified in Section 3.4 as a means to promoting hardware and device agnosticism of the probabilistic treatments presented. In practice, the capabilities or deficiencies of the underlying hardware unavoidably add variables to the analysis. For example, the VISAR system described in Chapter 6 exploits a fixed offset cursor due to the fact that the AR HMD tested upon does not currently support articulated finger tracking. The widely varying capabilities of currently available MR HMDs therefore inevitably impacts the true generalisability of the solutions identified.

8.6 Opportunities for Further Research

Chapters 4 to 7 each offer several avenues of future research related to the specific application and techniques examined in their respective case studies. This section itemises several more general opportunities for further research in the domain of probabilistic user interface design.

- **Full design cycle testing.** The emergent design process for probabilistic user interface design described in Chapter 2 has been demonstrated in part through the cases studies

examined in this thesis. Future work will seek to apply this process in a complete design cycle from idea formulation to system delivery.

- **Observational study with developers.** While this thesis offers several isolated demonstrations of the probabilistic user interface design approach, the appropriateness of these techniques for conventional MR application developers remains an open question. To this end, an observational study with developers who have received introductory training on these techniques would provide valuable insight into real-world utility of the approach.

- **Encode probabilistic treatments for practical use.** To promote developer uptake of the probabilistic user interface design approach and its associated techniques, there is likely value in some form of packaged encoding of these concepts. This may take the form of a design tool or set of prefabricated code blocks that could be added to a project. The most appropriate format requires investigation.

- **Refine the emergent design process.** The design process presented in Section 2.3 is reductive and ignores many of the complexities of system design. There is great value in examining how this process could be integrated within a more established and proven engineering design process as a means to accommodating these shortcomings. How a more complete design process could help guide probabilistic user interface design is interesting future work.

- **Isolate the need.** The framing of this thesis is based on several assumptions and qualitative observations regarding the unique challenges confronting the advancement of mixed reality interface design. These assumptions and observations are reasonable for motivating the work presented in this thesis but should be further investigated. For example, definitive work is required to understand what degree of input noise is to be expected in MR interactions or the extent to which the contextual adaptation of virtual content is required for user acceptance. Further research is required to properly isolate and document the unique requirements introduced in interface design for mixed reality in contrast to more established technologies.

- **Expand the system boundary.** The decision to draw the system boundary so as to ignore certain sources of uncertainty (e.g. the uncertain data produced by tracking or localisation sub-systems) is justified in Section 3.4. As the technology advances, however, it may be possible to model and communicate the qualities of these sub-systems so that they might be incorporated into the overarching probabilistic framework. Research work and demonstrative prototypes may be required to encourage device manufacturers to expose these signals.

8.7 Concluding Remarks

The emergence of low-cost consumer hardware offering compelling experiences in virtual and augmented reality paves the way for fundamentally new forms of work and leisure. The advancement of the underlying technology has, however, largely outstripped the application developer's ability to design and build effective interfaces and interactions for these environments. The challenging transition from passive experiences to truly interactive applications that seamlessly meld the physical and virtual worlds looms as a significant obstacle to the wider uptake of this technology. Designing next-generation applications in mixed reality that are interactive, engaging and satisfying to use is a difficult challenge. This thesis tests the hypothesis that probabilistic user interface design provides an effective methodology for delivering productive and enjoyable applications in mixed reality. The evaluation of this design approach is motivated by the assumption that an understanding of the uncertainty inherent to interaction and perception in mixed reality can be leveraged to make better design choices.

In testing this hypothesis, four research questions pertinent to understanding the potential of probabilistic user interface design in mixed reality are examined in four illustrative case studies: *characterisation*, *adaptation*, *inference* and *optimisation*. These case studies also provide the vehicle for demonstrating the versatility and range of techniques available under this approach, inline with the research objectives outlined in Section 1.3.2. Chapter 4 describes the *characterisation* of the probabilistic qualities of a text entry system for VR and illustrates how this understanding informs detailed design. This case study serves to address *Research Question 1* by demonstrating how simulating the behaviour of a functional input decoder can, with little development effort, enable the investigation of low-level user behaviours and performance. Section 4.6 describes how, for example, an input model for a fully functional decoder might be derived from the observed key targeting accuracy over the layout.

Chapter 5 presents an efficient strategy for resolving uncertainty around contextual dependence of virtual content in mixed reality by supporting *adaptation* via a probabilistic preference model. The data-driven methodology developed and demonstrated in response to *Research Question 2* highlights the efficiency of a crowdsourcing approach. The utility of the collected dataset to the application of dynamically adapting virtual content given the background setting is also demonstrated in Section 5.8.

Inference is applied in Chapter 6 to accommodate the high noise levels encountered in user input in a text entry system designed for AR. An error-tolerant text entry system leveraging an input decoder and a familiar touch-based interaction was developed as part of addressing *Research Question 3*. This system was shown to outperform a non-probabilistic state-of-the-art baseline and to deliver additional capabilities, such as the ability to type effectively with minimal field-of-view occlusion.

Finally, the unfamiliar and expansive design space for mixed reality is efficiently explored using a probabilistic *optimisation* approach paired with crowdsourcing in Chapter 7. The flexibility and efficiency of Bayesian optimisation serves as an illustrative solution to *Research Question 4*. The applicability of Bayesian optimisation to both familiar and completely novel interface and interaction design problems is demonstrated as part of the case study, recommending it as a worthy tool for the design of mixed reality applications. Together these four case studies demonstrate the potential of probabilistic user interface design in mixed reality.

In addition to the individual case studies offered, these concepts are unified within an emergent design process that is sketched out in Section 2.3. The four stages of this process are: i) characterise the user and the system; ii) isolate key determinants of performance; iii) examine sensitivity to design changes; and iv) refine and validate the system design. This design process is presented as a starting point upon which a more refined and complete process can be developed as part of future work. Nevertheless, the stages outlined in Section 2.3 provide useful preliminary guidance in applying structure to complex user interface design problems involving high levels of uncertainty. As a complete body of work, it is hoped that this thesis will contribute to the emerging design guidance for next-generation mixed reality applications.

References

[1] Ahn, E., Lee, S., and Kim, G. J. (2018). Real-time adjustment of contrast saliency for improved information visibility in mobile augmented reality. *Virtual Reality*, 22(3):245–262.

[2] Al-Sada, M., Ishizawa, F., Tsurukawa, J., and Nakajima, T. (2016). Input Forager: A User-driven Interaction Adaptation Approach for Head Worn Displays. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, MUM '16, pages 115–122, New York, NY, USA. ACM.

[3] Albarelli, A., Celentano, A., Cosmo, L., and Marchi, R. (2015). On the Interplay Between Data Overlay and Real-World Context Using See-through Displays. In *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*, CHItaly 2015, pages 58–65, New York, NY, USA. ACM.

[4] Amershi, S., Fogarty, J., and Weld, D. (2012). ReGroup: Interactive Machine Learning for On-Demand Group Creation in Social Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 21–30, New York, NY, USA. ACM.

[5] Arora, R., Kazi, R. H., Anderson, F., Grossman, T., Singh, K., and Fitzmaurice, G. (2017). Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 5643–5654, New York, NY, USA. ACM.

[6] Azenkot, S. and Zhai, S. (2012). Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, MobileHCI '12, pages 251–260, San Francisco, California, USA. Association for Computing Machinery.

[7] Bailly, G., Oulasvirta, A., Kötzing, T., and Hoppe, S. (2013). MenuOptimizer: Interactive Optimization of Menu Systems. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 331–342, New York, NY, USA. ACM.

[8] Bangor, A., Kortum, P. T., and Miller, J. T. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6):574–594.

[9] Bau, O. and Mackay, W. E. (2008). OctoPocus: A Dynamic Guide for Learning Gesture-based Command Sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 37–46, New York, NY, USA. ACM.

[10] Bi, X., Smith, B. A., and Zhai, S. (2012). Multilingual Touchscreen Keyboard Design and Optimization. *Human-Computer Interaction*, 27(4):352–382.

[11] Blessing, L. T. M. and Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. Springer-Verlag, London.

[12] Bohus, D. and Horvitz, E. (2009). Learning to Predict Engagement with a Spoken Dialog System in Open-world Settings. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '09, pages 244–252, Stroudsburg, PA, USA. Association for Computational Linguistics.

[13] Bornholt, J., Mytkowicz, T., and McKinley, K. S. (2014). Uncertain< T >: a first-order type for uncertain data. *ACM SIGPLAN Notices*, 49(4):51–66.

[14] Bowman, D. A., Rhoton, C. J., and Pinho, M. S. (2002). Text input techniques for immersive virtual environments: An empirical comparison. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 46, pages 2154–2158. SAGE Publications.

[15] Brochu, E., Brochu, T., and de Freitas, N. (2010). A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112. Eurographics Association.

[16] Brooke, J. (1996). SUS: A 'quick and dirty' usability scale. In *Usability Evaluation In Industry*, pages 189–194. CRC Press.

[17] Buschek, D. and Alt, F. (2017). ProbUI: Generalising Touch Target Representations to Enable Declarative Gesture Definition for Probabilistic GUIs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4640–4653, New York, NY, USA. ACM.

[18] Card, S. K., Moran, T. P., and Newell, A. (1983). *The psychology of human-computer interaction*. Lawrence Erlbaum Associates.

[19] Chai, J. Y., Hong, P., and Zhou, M. X. (2004). A Probabilistic Approach to Reference Resolution in Multimodal User Interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, IUI '04, pages 70–77, New York, NY, USA. ACM.

[20] Chen, X. A., Schwarz, J., Harrison, C., Mankoff, J., and Hudson, S. E. (2014). Air+Touch: Interweaving Touch & In-air Gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 519–525, New York, NY, USA. ACM.

[21] Choffin, B. and Ueda, N. (2018). Scaling Bayesian Optimization up to Higher Dimensions: a Review and Comparison of Recent Algorithms. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

[22] Chung, K., Ji, J. T., and So, R. H. Y. (2011). Manual control with time delays in an immersive virtual environment. In *Contemporary Ergonomics and Human Factors 2011: Proceedings of the international conference on Ergonomics & Human Factors 2011*, pages 211–218, Stoke Rochford, Lincolnshire. CRC Press.

[23] Clarkson, E., Clawson, J., Lyons, K., and Starner, T. (2005). An Empirical Study of Typing Rates on mini-QWERTY Keyboards. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1288–1291, New York, NY, USA. ACM.

[24] Clawson, J., Lyons, K., Rudnick, A., Iannucci, Jr., R. A., and Starner, T. (2008). Automatic Whiteout++: Correcting mini-QWERTY Typing Errors Using Keypress Timing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 573–582, New York, NY, USA. ACM.

[25] Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. (1997). QuickSet: Multimodal Interaction for Distributed Applications. In *Proceedings of the Fifth ACM International Conference on Multimedia*, MULTIMEDIA '97, pages 31–40, New York, NY, USA. ACM.

[26] Dahl, R., Norouzi, M., and Shlens, J. (2017). Pixel Recursive Super Resolution. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5449–5458. ISSN: 2380-7504.

[27] Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., and Allahbakhsh, M. (2018). Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *ACM Computing Surveys (CSUR)*, 51(1):7.

[28] David, P. A. (1985). Clio and the Economics of QWERTY. *The American economic review*, 75(2):332–337.

[29] Debernardis, S., Fiorentino, M., Gattullo, M., Monno, G., and Uva, A. E. (2014). Text Readability in Head-Worn Displays: Color and Style Optimization in Video versus Optical See-Through Devices. *IEEE Transactions on Visualization and Computer Graphics*, 20(1):125–139.

[30] Dhakal, V., Feit, A. M., Kristensson, P. O., and Oulasvirta, A. (2018). Observations on Typing from 136 Million Keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 646:1–646:12, New York, NY, USA. ACM.

[31] Doherty, G. J., Anderson, T., Wilson, M. D., and Faconti, G. P. (2001a). A control centred approach to designing interaction with novel devices. In *HCI*, pages 286–290.

[32] Doherty, G. J., Massink, M., and Faconti, G. (2001b). Reasoning About Interactive Systems with Stochastic Models. In *Proceedings of the 8th International Workshop on Interactive Systems: Design, Specification, and Verification-Revised Papers*, DSV-IS '01, pages 144–163, Berlin, Heidelberg. Springer-Verlag.

[33] Dudley, J. J., Benko, H., Wigdor, D., and Kristensson, P. O. (2019a). Performance Envelopes of Virtual Keyboard Text Input Strategies in Virtual Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality*, ISMAR 2019, pages 80–89. IEEEE.

[34] Dudley, J. J., Jacques, J. T., and Kristensson, P. O. (2019b). Crowdsourcing Interface Feature Design with Bayesian Optimization. In *Proceedings of the 2019 Conference on Human Factors in Computing Systems*, CHI '19, pages 252:1–252:12, New York, NY, USA. ACM.

[35] Dudley, J. J., Jacques, J. T., and Kristensson, P. O. (2020). Crowdsourcing Design Guidance for Augmented Reality: A Use Case in Contextually-Adaptive Text Content. *Under Submission*.

[36] Dudley, J. J. and Kristensson, P. O. (2018). A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):8:1–8:37.

[37] Dudley, J. J., Schuff, H., and Kristensson, P. O. (2018a). Bare-Handed *3D* Drawing in Augmented Reality. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, pages 241–252, New York, NY, USA. ACM.

[38] Dudley, J. J., Vertanen, K., and Kristensson, P. O. (2018b). Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(6):30:1–30:40.

[39] Dumas, B., Signer, B., and Lalanne, D. (2012). Fusion in Multimodal Interactive Systems: An HMM-based Algorithm for User-induced Adaptation. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '12, pages 15–24, New York, NY, USA. ACM.

[40] Duvenaud, D. (2014). *Automatic model construction with Gaussian processes*. Thesis, University of Cambridge.

[41] Feit, A. M., Weir, D., and Oulasvirta, A. (2016). How We Type: Movement Strategies and Performance in Everyday Typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4262–4273, New York, NY, USA. ACM.

[42] Findlater, L. and Wobbrock, J. (2012). Personalized Input: Improving Ten-finger Touchscreen Typing Through Automatic Adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 815–824, New York, NY, USA. ACM.

[43] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391.

[44] Fogarty, J., Tan, D., Kapoor, A., and Winder, S. (2008). CueFlik: Interactive Concept Learning in Image Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 29–38, New York, NY, USA. ACM.

[45] Fogg, B. J. and Tseng, H. (1999). The Elements of Computer Credibility. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 80–87, New York, NY, USA. ACM.

[46] Gabbard, J. L. (2008). *Usability Engineering of Text Drawing Styles in Augmented Reality User Interfaces*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

[47] Gabbard, J. L. and Swan, J. E. (2008). Usability Engineering for Augmented Reality: Employing User-Based Studies to Inform Design. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):513–525.

[48] Gabbard, J. L., Swan, J. E., Hix, D., Schulman, R. S., Lucas, J., and Gupta, D. (2005). An empirical user-based study of text drawing styles and outdoor background textures for augmented reality. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pages 11–18.

[49] Gadiraju, U., Kawase, R., Dietze, S., and Demartini, G. (2015). Understanding Malicious Behavior in Crowdsourcing Platforms: The Case of Online Surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1631–1640, New York, NY, USA. ACM.

[50] Gajos, K. and Weld, D. S. (2004). SUPPLE: Automatically Generating User Interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, IUI '04, pages 93–100, New York, NY, USA. ACM.

[51] Gillies, M., Brenton, H., and Kleinsmith, A. (2015). Embodied Design of Full Bodied Interaction with Virtual Humans. In *Proceedings of the 2nd International Workshop on Movement and Computing*, MOCO '15, pages 1–8, New York, NY, USA. ACM.

[52] González, J., Dai, Z., Hennig, P., and Lawrence, N. (2016). Batch Bayesian Optimization via Local Penalization. In Gretton, A. and Robert, C. C., editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 648–657, Cadiz, Spain. PMLR.

[53] Goodman, J., Venolia, G., Steury, K., and Parker, C. (2002). Language Modeling for Soft Keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 194–195, New York, NY, USA. ACM.

[54] Greis, M., Karolus, J., Schuff, H., Woźniak, P., and Henze, N. (2017). Detecting Uncertain Input Using Physiological Sensing and Behavioral Measurements. In *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*, MUM '17, pages 299–304, New York, NY, USA. ACM.

[55] Grossman, T., Chen, X. A., and Fitzmaurice, G. (2015). Typing on Glasses: Adapting Text Entry to Smart Eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages 144–152, New York, NY, USA. ACM.

[56] Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., and Kristensson, P. O. (2018a). Effects of Hand Representations for Typing in Virtual Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 151–158.

[57] Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., and Kristensson, P. O. (2018b). Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 159–166.

[58] Hagiya, T. and Kato, T. (2012). Probabilistic keyboard adaptable to user and operating style based on syllable HMMs. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 65–68.

[59] Hagiya, T. and Kato, T. (2013). Adaptable Probabilistic Flick Keyboard Based on HMMs. In *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion*, IUI '13 Companion, pages 71–72, New York, NY, USA. ACM.

[60] Hanson, R., Falkenström, W., and Miettinen, M. (2017). Augmented reality as a means of conveying picking information in kit preparation for mixed-model assembly. *Computers & Industrial Engineering*, 113:570 – 575.

[61] Harmon, R., Patterson, W., Ribarsky, W., and Bolter, J. (1996). The virtual annotation system. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 239–245. IEEE.

[62] Hasler, D. and Suesstrunk, S. E. (2003). Measuring colorfulness in natural images. In *Human Vision and Electronic Imaging VIII*, volume 5007, pages 87–95. International Society for Optics and Photonics.

[63] Heer, J. and Bostock, M. (2010). Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 203–212, New York, NY, USA. ACM.

[64] Hick, W. E. (1952). On the rate of gain of information. *The Quarterly Journal of Experimental Psychology*, 4:11–26.

[65] Hincapié-Ramos, J. D., Guo, X., Moghadasian, P., and Irani, P. (2014). Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1063–1072. ACM.

[66] Hincapié-Ramos, J. D., Ivanchuk, L., Sridharan, S. K., and Irani, P. P. (2015). SmartColor: Real-Time Color and Contrast Correction for Optical See-Through Head-Mounted Displays. *IEEE Transactions on Visualization and Computer Graphics*, 21(12):1336–1348.

[67] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and others (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[68] Hoffmann, E. R. (1992). Fitts' Law with transmission delay. *Ergonomics*, 35(1):37–48.

[69] Horvitz, E. (1999). Principles of Mixed-initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 159–166, New York, NY, USA. ACM.

[70] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The LumièRe Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 256–265, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[71] Hoste, L. and Signer, B. (2013). SpeeG2: A Speech- and Gesture-based Interface for Efficient Controller-free Text Input. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 213–220, New York, NY, USA. ACM.

[72] Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45(3):188–196.

[73] Jacques, J. T. and Kristensson, P. O. (2013). Crowdsourcing a hit: measuring workers' pre-task interactions on microtask markets. In *First AAAI Conference on Human Computation and Crowdsourcing*.

[74] Jacques, J. T. and Kristensson, P. O. (2017). Design Strategies for Efficient Access to Mobile Device Users via Amazon Mechanical Turk. In *Proceedings of the First ACM Workshop on Mobile Crowdsensing Systems and Applications*, CrowdSenSys '17, pages 25–30, New York, NY, USA. ACM. Delft, Netherlands.

[75] Kang, R., Brown, S., Dabbish, L., and Kiesler, S. (2014). Privacy attitudes of Mechanical Turk workers and the US public. In *Symposium on Usable Privacy and Security (SOUPS)*, volume 4, page 1.

[76] Kapoor, A., Lee, B., Tan, D., and Horvitz, E. (2010). Interactive Optimization for Steering Machine Classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1343–1352, New York, NY, USA. ACM.

[77] Khajah, M. M., Roads, B. D., Lindsey, R. V., Liu, Y.-E., and Mozer, M. C. (2016). Designing Engaging Games Using Bayesian Optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5571–5582, New York, NY, USA. ACM.

[78] Kim, B., Glassman, E., Johnson, B., and Shah, J. (2015). iBCM: Interactive Bayesian Case Model Empowering Humans via Intuitive Interaction. Technical report, MIT Computer Science and Artificial Intelligence Laboratory.

[79] Kim, S., Sohn, M., Pak, J., and Lee, W. (2006). One-key Keyboard: A Very Small QWERTY Keyboard Supporting Text Entry for Wearable Computing. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, OZCHI '06, pages 305–308, New York, NY, USA. ACM.

[80] Kin, K., Agrawala, M., and DeRose, T. (2009). Determining the Benefits of Direct-touch, Bimanual, and Multifinger Input on a Multitouch Workstation. In *Proceedings of Graphics Interface 2009*, GI '09, pages 119–124, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.

[81] Kirwan, B. and Ainsworth, L. K. (1992). *A guide to task analysis: the task analysis working group*. CRC press.

[82] Kittur, A., Chi, E. H., and Suh, B. (2008). Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA. ACM.

[83] Knierim, P., Schwind, V., Feit, A. M., Nieuwenhuizen, F., and Henze, N. (2018). Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 345:1–345:9, New York, NY, USA. ACM.

[84] Kohavi, R., Henne, R. M., and Sommerfield, D. (2007). Practical Guide to Controlled Experiments on the Web: Listen to Your Customers Not to the Hippo. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 959–967, New York, NY, USA. ACM.

[85] Kohli, L. and Whitton, M. (2005). The Haptic Hand: Providing User Interface Feedback with the Non-dominant Hand in Virtual Environments. In *Proceedings of Graphics Interface 2005*, GI '05, pages 1–8, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada. Canadian Human-Computer Communications Society.

[86] Komarov, S., Reinecke, K., and Gajos, K. Z. (2013). Crowdsourcing Performance Evaluations of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 207–216, New York, NY, USA. ACM.

[87] Koyama, Y., Sato, I., Sakamoto, D., and Igarashi, T. (2017). Sequential Line Search for Efficient Visual Design Optimization by Crowds. *ACM Trans. Graph.*, 36(4):48:1–48:11.

[88] Kristensson, P. O. (2015). Next-Generation Text Entry. *Computer*, 48(7):84–87.

[89] Kristensson, P.-O. and Zhai, S. (2004). SHARK 2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 43–52. ACM.

[90] Kristensson, P. O. and Zhai, S. (2005). Relaxing Stylus Typing Precision by Geometric Pattern Matching. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, pages 151–158, New York, NY, USA. ACM.

[91] Kruijff, E., Orlosky, J., Kishishita, N., Trepkowski, C., and Kiyokawa, K. (2018). The Influence of Label Design on Search Performance and Noticeability in Wide Field of View Augmented Reality Displays. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1.

[92] Kruijff, E., Swan, J. E., and Feiner, S. (2010). Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12.

[93] Kuester, F., Chen, M., Phair, M. E., and Mehring, C. (2005). Towards Keyboard Independent Touch Typing in VR. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '05, pages 86–95, New York, NY, USA. ACM.

[94] Kulesza, T., Stumpf, S., Burnett, M., Yang, S., Kwan, I., and Wong, W.-K. (2013). Too much, too little, or just right? Ways explanations impact end users' mental models. In *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*, pages 3–10.

[95] Lasecki, W. S., Gordon, M., Leung, W., Lim, E., Bigham, J. P., and Dow, S. P. (2015). Exploring Privacy and Accuracy Trade-Offs in Crowdsourced Behavioral Video Coding. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1945–1954, New York, NY, USA. Association for Computing Machinery.

[96] Lasecki, W. S., Song, Y. C., Kautz, H., and Bigham, J. P. (2013). Real-Time Crowd Labeling for Deployable Activity Recognition. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 1203–1212, New York, NY, USA. Association for Computing Machinery.

[97] Le, H. V., Mayer, S., and Henze, N. (2019). Investigating the Feasibility of Finger Identification on Capacitive Touchscreens Using Deep Learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 637–649, New York, NY, USA. ACM.

[98] Lee, M. and Woo, W. (2003). ARKB: 3D vision-based Augmented Reality Keyboard. In *ICAT*.

[99] Leykin, A. and Tuceryan, M. (2004). Automatic determination of text readability over textured backgrounds for augmented reality systems. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 224–230.

[100] Liebowitz, S. J. and Margolis, S. E. (1990). The fable of the keys. *The journal of law & economics*, 33(1):1–25.

[101] Lindeman, R. W. (1999). *Bimanual Interaction, Passive-haptic Feedback, 3D Widget Representation, and Simulated Surface Constraints for Interaction in Immersive Virtual Environments*. PhD Thesis, The George Washington University.

[102] Liu, J., Wong, C. K., and Hui, K. K. (2003). An adaptive user interface based on personalized learning. *IEEE Intelligent Systems*, 18(2):52–57.

[103] Liu, W., D'Oliveira, R. L., Beaudouin-Lafon, M., and Rioul, O. (2017). BIGnav: Bayesian Information Gain for Guiding Multiscale Navigation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 5869–5880, New York, NY, USA. ACM.

[104] Liu, W., Rioul, O., McGrenere, J., Mackay, W. E., and Beaudouin-Lafon, M. (2018). BIGFile: Bayesian Information Gain for Fast File Retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 385:1–385:13, New York, NY, USA. ACM.

[105] Liu, Y.-E., Mandel, T., Brunskill, E., and Popović, Z. (2014). Towards Automatic Experimentation of Educational Knowledge. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3349–3358, New York, NY, USA. ACM.

[106] Livingston, M. A., Gabbard, J. L., Swan, J. E., Sibley, C. M., and Barrow, J. H. (2013). Basic Perception in Head-Worn Augmented Reality Displays. In Huang, W., Alem, L., and Livingston, M. A., editors, *Human Factors in Augmented Reality Environments*, pages 35–65. Springer New York, New York, NY.

[107] Lizotte, D. J. (2008). *Practical Bayesian Optimization*. PhD Thesis, University of Alberta, Edmonton, Canada.

[108] Lomas, J. D., Forlizzi, J., Poonwala, N., Patel, N., Shodhan, S., Patel, K., Koedinger, K., and Brunskill, E. (2016). Interface Design Optimization As a Multi-Armed Bandit Problem. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4142–4153, New York, NY, USA. ACM.

[109] Long, J. and Dowell, J. (1989). Conceptions of the Discipline of HCI: Craft, Applied Science, and Engineering. In *People and Computers V: Proceedings of the Fifth Conference of the British Computer Society*, volume 5, page 9. Cambridge University Press.

[110] Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., and Looney, E. W. (2004). Twiddler Typing: One-handed Chording Text Entry for Mobile Phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 671–678, New York, NY, USA. ACM.

[111] Mahmud, M. H., Rosman, B., Ramamoorthy, S., and Kohli, P. (2014). Adapting interaction environments to diverse users through online action set selection. In *Proceedings of the AAAI 2014 Workshop on Machine Learning for Interactive Systems*. Citeseer.

[112] Manghisi, V. M., Gattullo, M., Fiorentino, M., Uva, A. E., Marino, F., Bevilacqua, V., and Monno, G. (2017). Predicting Text Legibility over Textured Digital Backgrounds for a Monocular Optical See-Through Display. *Presence: Teleoperators and Virtual Environments*, 26(1):1–15.

[113] Mankoff, J., Hudson, S. E., and Abowd, G. D. (2000). Interaction Techniques for Ambiguity Resolution in Recognition-based Interfaces. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 11–20, New York, NY, USA. ACM.

[114] Markussen, A., Jakobsen, M. R., and Hornbaek, K. (2013). Selection-Based Mid-Air Text Entry on Large Displays. In Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., and Winckler, M., editors, *Human-Computer Interaction – INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part I*, pages 401–418. Springer Berlin Heidelberg, Berlin, Heidelberg.

[115] Markussen, A., Jakobsen, M. R., and Hornbaek, K. (2014). Vulture: A Mid-air Word-gesture Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1073–1082, New York, NY, USA. ACM.

[116] McDuff, D., Kaliouby, R. E., and Picard, R. W. (2012). Crowdsourcing Facial Responses to Online Videos. *IEEE Transactions on Affective Computing*, 3(4):456–468.

[117] McGrath, J. E. (1981). Dilemmatics: The study of research choices and dilemmas. *American Behavioral Scientist*, 25(2):179–210.

[118] McGraw, T., Garcia, E., and Sumner, D. (2017). Interactive Swept Surface Modeling in Virtual Reality with Motion-tracked Controllers. In *Proceedings of the Symposium on Sketch-Based Interfaces and Modeling*, SBIM '17, pages 4:1–4:9, New York, NY, USA. ACM.

[119] McRuer, D. T. and Krendel, E. S. (1974). Mathematical Models of Human Pilot Behavior. Technical Report AGARD-AG-188, Advisory Group for Aerospace Research and Development, Neuilly-Sur-Seine, France.

[120] Micallef, L., Dragicevic, P., and Fekete, J.-D. (2012). Assessing the effect of visualizations on bayesian reasoning through crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2536–2545.

[121] Micallef, L., Palmas, G., Oulasvirta, A., and Weinkauf, T. (2017). Towards Perceptual Optimization of the Visual Design of Scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1588–1599.

[122] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329.

[123] Montague, K., Hanson, V. L., and Cobley, A. (2012). Designing for Individuals: Usable Touch-screen Interaction Through Shared User Models. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '12, pages 151–158, New York, NY, USA. ACM.

[124] Mott, M. E. and Wobbrock, J. O. (2019). Cluster Touch: Improving Touch Accuracy on Smartphones for People with Motor and Situational Impairments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 27:1–27:14, New York, NY, USA. ACM.

[125] Muir, B. M. and Moray, N. (1996). Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics*, 39(3):429–460.

[126] Myung, J. I., Cavagnaro, D. R., and Pitt, M. A. (2013). A Tutorial on Adaptive Design Optimization. *Journal of Mathematical Psychology*, 57(3):53 – 67.

[127] Newton, E. M., Sweeney, L., and Malin, B. (2005). Preserving privacy by de-identifying face images. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):232–243.

[128] Ni, T., Bowman, D., and North, C. (2011). AirStroke: Bringing Unistroke Text Entry to Freehand Gesture Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2473–2476, New York, NY, USA. ACM.

[129] Nielsen, J. B. B., Nielsen, J., and Larsen, J. (2015). Perception-Based Personalization of Hearing Aids Using Gaussian Processes and Active Learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):162–173.

[130] Norman, D. A. (2014). Some Observations on Mental Models. In Gentner, D. and Stevens, A. L., editors, *Mental Models*, pages 7–14. Psychology Press, New York.

[131] Octavia, J. R., Raymaekers, C., and Coninx, K. (2011). Adaptation in virtual environments: conceptual framework and user models. *Multimedia Tools and Applications*, 54(1):121–142.

[132] O'Donovan, P., Agarwala, A., and Hertzmann, A. (2011). Color Compatibility from Large Datasets. *ACM Trans. Graph.*, 30(4):63:1–63:12.

[133] O'Donovan, P., Agarwala, A., and Hertzmann, A. (2015). DesignScape: Design with Interactive Layout Suggestions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1221–1224, New York, NY, USA. ACM.

[134] Ogitani, T., Arahori, Y., Shinyama, Y., and Gondow, K. (2018). Space Saving Text Input Method for Head Mounted Display with Virtual 12-key Keyboard. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 342–349.

[135] Orlosky, J., Kiyokawa, K., and Takemura, H. (2013). Dynamic Text Management for See-through Wearable and Heads-up Display Systems. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 363–370, New York, NY, USA. ACM.

[136] Orlosky, J., Kiyokawa, K., and Takemura, H. (2014). Managing Mobile Text in Head Mounted Displays: Studies on Visual Preference and Text Placement. *SIGMOBILE Mob. Comput. Commun. Rev.*, 18(2):20–31.

[137] Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. (2005). Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 919–928. ACM.

[138] Ovaska, S. and Räihä, K.-J. (2009). Teaching Privacy with Ubicomp Scenarios in HCI Classes. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, OZCHI '09, pages 105–112, New York, NY, USA. ACM.

[139] Paavilainen, J., Korhonen, H., Alha, K., Stenros, J., Koskinen, E., and Mayra, F. (2017). The PokéMon GO Experience: A Location-Based Augmented Reality Mobile Game Goes Mainstream. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2493–2498, New York, NY, USA. ACM.

[140] Pahl, G. and Beitz, W. (2013). *Engineering design: a systematic approach*. Springer Science & Business Media.

[141] Pang, A. T., Wittenbrink, C. M., and Lodha, S. K. (1997). Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390.

[142] Park, S., Gebhardt, C., Rädle, R., Feit, A. M., Vrzakova, H., Dayama, N. R., Yeo, H.-S., Klokmose, C. N., Quigley, A., Oulasvirta, A., and Hilliges, O. (2018). AdaM: Adapting Multi-User Interfaces for Collaborative Environments in Real-Time. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 184:1–184:14, New York, NY, USA. ACM.

[143] Payne, S. J. and Howes, A. (2013). *Adaptive Interaction: A Utility Maximization Approach to Understanding Human Interaction with Technology*. Morgan & Claypool Publishers, 1st edition.

[144] Pearl, J. (2014). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier. Google-Books-ID: mn2jBQAAQBAJ.

[145] Pick, S., Puika, A. S., and Kuhlen, T. W. (2016). SWIFTER: Design and evaluation of a speech-based text input metaphor for immersive virtual environments. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 109–112. IEEE.

[146] Poupyrev, I., Tomokazu, N., and Weghorst, S. (1998). Virtual Notepad: handwriting in immersive VR. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pages 126–132. IEEE.

[147] Prätorius, M., Valkov, D., Burgbacher, U., and Hinrichs, K. (2014). DigiTap: an eyes-free VR/AR symbolic input device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 9–18. ACM.

[148] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

[149] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA. ACM.

[150] Ridpath, C. and Chisholm, W. (2000). *Techniques For Accessibility Evaluation And Repair Tools*.

[151] Rogers, S., Williamson, J., Stewart, C., and Murray-Smith, R. (2010). FingerCloud: Uncertainty and Autonomy Handover Incapacitive Sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 577–580, New York, NY, USA. ACM.

[152] Rogers, S., Williamson, J., Stewart, C., and Murray-Smith, R. (2011). AnglePose: Robust, Precise Capacitive Touch Tracking via 3D Orientation Estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2575–2584, New York, NY, USA. ACM.

[153] Rosenberg, R. and Slater, M. (1999). The chording glove: a glove-based text input device. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 29(2):186–191.

[154] Ruan, S., Wobbrock, J. O., Liou, K., Ng, A., and Landay, J. A. (2018). Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4):159:1–159:23.

[155] Sacha, D., Senaratne, H., Kwon, B. C., Ellis, G., and Keim, D. A. (2016). The role of uncertainty, awareness, and trust in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):240–249.

[156] Sajjadi, M. S. M., Schölkopf, B., and Hirsch, M. (2017). EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4501–4510. ISSN: 2380-7504.

[157] Salem, P. (2017). User Interface Optimization Using Genetic Programming with an Application to Landing Pages. *Proc. ACM Hum.-Comput. Interact.*, 1(EICS):13:1–13:17.

[158] Samuel, A. and Weir, J. (1999). *Introduction to Engineering Design*. Elsevier.

[159] Sarcar, S., Joklnen, J., Oulasvirta, A., Silpasuwanchai, C., Wang, Z., and Ren, X. (2016). Towards Ability-Based Optimization for Aging Users. In *Proceedings of the International Symposium on Interactive Technology and Ageing Populations*, ITAP '16, pages 77–86, New York, NY, USA. ACM.

[160] Sarkar, A. (2015). Confidence, command, complexity: metamodels for structured interaction with machine intelligence. In *Proceedings of the 26th Annual Conference of the Psychology of Programming Interest Group (PPIG 2015)*, pages 23–36.

[161] Sarkar, A., Jamnik, M., Blackwell, A. F., and Spott, M. (2015). Interactive visual machine learning in spreadsheets. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 159–163.

[162] Schwarz, J. (2015). *Monte Carlo Methods for Managing Uncertain User Interfaces*. PhD thesis, Carnegie Mellon University.

[163] Schwarz, J., Hudson, S., Mankoff, J., and Wilson, A. D. (2010). A Framework for Robust and Flexible Handling of Inputs with Uncertainty. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 47–56, New York, NY, USA. ACM.

[164] Schwarz, J., Mankoff, J., and Hudson, S. (2011). Monte Carlo Methods for Managing Interactive State, Action and Feedback Under Uncertainty. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 235–244, New York, NY, USA. ACM.

[165] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and Freitas, N. d. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175.

[166] Sheridan, T. B. (1992). Musings on Telepresence and Virtual Presence. *Presence: Teleoperators and Virtual Environments*, 1(1):120–126.

[167] Sheridan, T. B. and Ferrell, W. R. (1974). *Man-machine systems; Information, control, and decision models of human performance*. The MIT Press, Cambridge, MA, US.

[168] Shi, W., Yu, C., Yi, X., Li, Z., and Shi, Y. (2018). TOAST: Ten-Finger Eyes-Free Typing on Touchable Surfaces. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1):33:1–33:23.

[169] Snoek, J. (2013). *Bayesian Optimization and Semiparametric Models with Applications to Assistive Technology*. PhD Thesis, University of Toronto, Toronto, Canada.

[170] Sommerville, I. (2010). *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition.

[171] Sridhar, S., Feit, A. M., Theobalt, C., and Oulasvirta, A. (2015). Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3643–3652, New York, NY, USA. ACM.

[172] Stockman, G. and Shapiro, L. G. (2001). *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.

[173] Stolcke, A. (2002). SRILM – An Extensible Language Modeling Toolkit. In *International Conference on Spoken Language Processing*, pages 901–904.

[174] Tan, C. T., Sapkota, H., and Rosser, D. (2014). BeFaced: A Casual Game to Crowdsource Facial Expressions in the Wild. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 491–494, New York, NY, USA. ACM.

[175] Tanaka, K., Kishino, Y., Miyamae, M., Terada, T., and Nishio, S. (2007). An Information Layout Method for an Optical See-through HMD Considering the Background. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 109–110. ISSN: 2376-8541.

[176] Tanaka, K., Kishino, Y., Miyamae, M., Terada, T., and Nishio, S. (2008). An Information Layout Method for an Optical See-through Head Mounted Display Focusing on the Viewability. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 139–142, Washington, DC, USA. IEEE Computer Society.

[177] Todi, K., Weir, D., and Oulasvirta, A. (2016). Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, pages 543–555, New York, NY, USA. ACM.

[178] Toomim, M., Kriplean, T., Pörtner, C., and Landay, J. (2011). Utility of Human-computer Interactions: Toward a Science of Preference Measurement. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2275–2284, New York, NY, USA. ACM.

[179] Vad, B., Boland, D., Williamson, J., Murray-Smith, R., and Steffensen, P. B. (2015). Design and Evaluation of a Probabilistic Music Projection Interface.

[180] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

[181] Verlinden, J. C., Bolter, J. D., and van der Mast, C. (1993). Virtual Annotation: Verbal Communication in Virtual Reality. Technical Report GIT-GVU-93-40, Georgia Institute of Technology.

[182] Vertanen, K., Gaines, D., Fletcher, C., Stanage, A. M., Watling, R., and Kristensson, P. O. (2019). VelociWatch: Designing and Evaluating a Virtual Keyboard for the Input of Challenging Text. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 591:1–591:14, New York, NY, USA. ACM.

[183] Vertanen, K. and Kristensson, P. O. (2008). On the Benefits of Confidence Visualization in Speech Recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1497–1500, New York, NY, USA. ACM.

[184] Vertanen, K. and Kristensson, P. O. (2009). Parakeet: A Continuous Speech Recognition System for Mobile Touch-screen Devices. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 237–246, New York, NY, USA. ACM.

[185] Vertanen, K. and Kristensson, P. O. (2011). A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 295–298. ACM.

[186] Vertanen, K. and Kristensson, P. O. (2014). Complementing text entry evaluations with a composition task. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(2):8.

[187] Vertanen, K., Memmi, H., Emge, J., Reyal, S., and Kristensson, P. O. (2015). VelociTap: investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 659–668. ACM.

[188] Vidulin, V., Bohanec, M., and Gams, M. (2014). Combining human analysis and machine data mining to obtain credible data relations. *Information Sciences*, 288:254–278.

[189] Walker, J., Li, B., Vertanen, K., and Kuhl, S. (2017). Efficient Typing on a Visually Occluded Physical Keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 5457–5461, New York, NY, USA. ACM.

[190] Wang, C.-Y., Chu, W.-C., Chiu, P.-T., Hsiu, M.-C., Chiang, Y.-H., and Chen, M. Y. (2015). PalmType: Using Palms As Keyboards for Smart Glasses. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages 153–160, New York, NY, USA. ACM.

[191] Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and De Freitas, N. (2016). Bayesian Optimization in a Billion Dimensions via Random Embeddings. *Journal of Artificial Intelligence Research*, 55:361–387.

[192] Weir, D., Pohl, H., Rogers, S., Vertanen, K., and Kristensson, P. O. (2014). Uncertain Text Entry on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2307–2316, New York, NY, USA. ACM.

[193] Weir, D., Rogers, S., Murray-Smith, R., and Löchtefeld, M. (2012). A User-specific Machine Learning Approach for Improving Touch Accuracy on Mobile Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 465–476, New York, NY, USA. ACM.

[194] Williamson, J. (2006). *Continuous uncertain interaction*. University of Glasgow (United Kingdom).

[195] Williamson, J. and Murray-Smith, R. (2005). Hex: Dynamics and Probabilistic Text Entry. In Murray-Smith, R. and Shorten, R., editors, *Switching and Learning in Feedback Systems: European Summer School on Multi-Agent Control, Maynooth, Ireland, September 8-10, 2003, Revised Lectures and Selected Papers*, Lecture Notes in Computer Science, pages 333–342. Springer Berlin Heidelberg, Berlin, Heidelberg.

[196] Wilson, A. and Shafer, S. (2003). XWand: UI for Intelligent Spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 545–552, New York, NY, USA. ACM.

[197] Wolf, D., Dudley, J. J., and Kristensson, P. O. (2018). Performance Envelopes of in-Air Direct and Smartwatch Indirect Control for Head-Mounted Augmented Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 347–354. IEEE.

[198] Yi, X., Yu, C., Zhang, M., Gao, S., Sun, K., and Shi, Y. (2015). ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3D Hand Tracking Data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 539–548, New York, NY, USA. ACM.

[199] Yu, C., Gu, Y., Yang, Z., Yi, X., Luo, H., and Shi, Y. (2017). Tap, Dwell or Gesture?: Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4479–4488, New York, NY, USA. ACM.

[200] Yu, C., Sun, K., Zhong, M., Li, X., Zhao, P., and Shi, Y. (2016). One-Dimensional Handwriting: Inputting Letters and Words on Smart Glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 71–82. ACM.

[201] Yu, D., Fan, K., Zhang, H., Monteiro, D., Xu, W., and Liang, H. (2018). PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2927–2935.

[202] Zhai, S. and Kristensson, P. O. (2012). The Word-gesture Keyboard: Reimagining Keyboard Interaction. *Communications of the ACM*, 55(9):91–101.

[203] Zhai, S., Kristensson, P. O., and Smith, B. A. (2005). In search of effective text input interfaces for off the desktop computing. *Interacting with computers*, 17(3):229–250.